THESE

présentée

A LA FACULTÉ DES SCIENCES DE PARIS VI

pour obtenir

LE GRADE DE DOCTEUR-INGÉNIEUR

SPÉCIALITÉ : MATHÉMATIQUES

MENTION : ANALYSE NUMÉRIQUE

par

Jean-Pierre QUADRAT

MÉTHODES DE SIMULATION EN PROGRAMMATION DYNAMIQUE STOCHASTIQUE

Soutenue le 15 Juin 1973 devant la Commission d'examen composée de :

MM. J.L. LIONS — *Président*

A. BENSOUSSAN, R. GLOWINSKI — *Examinateurs*

THESE

présentée

A LA FACULTÉ DES SCIENCES DE PARIS VI

pour obtenir

LE GRADE DE DOCTEUR-INGÉNIEUR

SPÉCIALITÉ : MATHÉMATIQUES

MENTION : ANALYSE NUMÉRIQUE

par

Jean-Pierre QUADRAT

MÉTHODES DE SIMULATION EN PROGRAMMATION DYNAMIQUE STOCHASTIQUE

Soutenue le 15 Juin 1973 devant la Commission d'examen composée de :

MM. J.L. LIONS — *Président*

A. BENSOUSSAN, R. GLOWINSKI — *Examinateurs*

UNIV.
LEIDEN

Je remercie Monsieur LIONS pour m'avoir donné les possibilités matérielles de réaliser ce travail ; Monsieur BENSOUSSAN qui m'a fourni le sujet, constamment aidé par ses conseils et ses encouragements, ainsi que Monsieur GLOWINSKI pour avoir accepté de participer au jury.

Je remercie tous mes camarades de l'I.R.I.A., et plus particulièrement Monsieur VIOT avec qui j'ai eu la chance d'étudier le théorème de convergence (1ere partie) et qui m'a beaucoup appris.

Je remercie également Mademoiselle GRANGEARD pour le soin avec lequel elle a réalisé ce mémoire.

INTRODUCTION

Un certain nombre d'algorithmes résolvent des problèmes de programmation dynamique stochastique la méthode de KALMAN, l'algorithme de HOWARD, l'utilisation du principe de BELLMAN et un grand nombre de variantes.

La première permet de résoudre le cas où le système est linéaire, le coût quadratique et le bruit gaussien. Les deux autres sont très limités par la dimension de l'état du système.

Nous essayions de donner ici des algorithmes permettant de résoudre une classe plus large de problèmes, bien que souvent, d'une façon approchée.

JACOBSON et MAYNE (11) donnent un algorithme d'optimisation qui, à partir de simulations de trajectoires du bruit du système, donne la solution open loop.

Nous avons repris cette idée :

- Nous simulons des trajectoires du bruit du système,
- Nous construisons sur ces réalisations un problème déterministe,
- Nous résolvons ce dernier problème par des méthodes déterministes classiques,
- Nous montrons que la solution ainsi obtenue converge vers la solution du problème initial lorsque le nombre de réalisations du bruit augmente.

Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41552718_0003

(Nous obtenons ainsi des solutions open loop ou feedback selon la classe de stratégies dans laquelle on minimise).

La première partie étudie la convergence des problèmes déterministes construits sur les réalisations du bruit du système, vers le problème initial. Elle est le résultat d'un travail commun avec M. VIOT.

La seconde partie explicite les problèmes déterministes construits sur les réalisations du bruit dans le cas de la recherche d'une solution open loop ou Feedback. Nous approchons également le problème Feedback par un problème que nous appelons relaxé et qui a l'avantage d'être plus maniable numériquement.

Nous rappelons les divers algorithmes déterministes classiques qui nous permettent de résoudre nos problèmes.

Nous donnons également rapidement les méthodes usuelles de résolution des problèmes de programmation dynamique stochastique qui nous serviront de test.

Nous rappelons quelques façons de générer numériquement des variables aléatoires.

Nous montrons également comment il est possible de traiter les contraintes sur l'état en probabilité bien que de façon très couteuse.

Dans la troisième partie nous appliquons nos résultats sur des exemples :

- Un exemple simple sur lequel on peut utiliser la méthode de Kalman.

- Un exemple de gestion stock-production d'un produit qui nous permet de comparer nos algorithmes à la programmation dynamique classique.

- Une gestion de stock-production de 5 produits et une ressource où nous utilisons nos algorithmes en temps réel. (Cet exemple ne peut pas être traîté directement par les autres méthodes).

Dans ces exemples nous essayons de montrer l'intérêt et les limites de nos méthodes.

PLAN

I - METHODES DE SIMULATION EN PROGRAMMATION DYNAMIQUE STOCHASTIQUE.

§ 1. Systèmes et stratégies.

§ 2. Quelques propriétés de la valeur Φ.

§ 3. Loi forte des grands nombres en programmation dynamique stochastique;

1. Rappels sur la loi forte des grands nombres.

2. Application à la simulation indépendante.

3. Cas des contraintes presque-sûres.

II - ALGORITHMES DE RESOLUTIONS DES PROBLEMES $P(\mu_r)$.

§ 1. Les problèmes P_r, R_r, O_r .

1. La classe $\mathcal{S}_k$ de stratégies.

2. Le problème P_r .

3. Le problème R_r .

4. Le problème O_r .

§ 2. Algorithmes de résolution des problèmes P_r, O_r, R_r.

1. Pénalisation extérieure.

2. Méthodes duales

2.2.1. Dualisation des contraintes sur l'état.

2.2.2. Décomposition.

3. Algorithme de gradient.

4. Algorithme d'UZAWA.

5. Calcul des gradients des problèmes P_r, R_r, O_r.

5.1. Calcul du gradient pour un problème de contrôle à temps discret.

5.2. Application au problème P_r pénalisé.

5.3. Application au problème R_r pénalisé.

§ 3. Les méthodes de référence.

1. La programmation dynamique classique.

2. La méthode de KALMAN.

§ 4. Le problème $P(\mu_r,\alpha_r)$

§ 5. Génération de réalisation de variables aléatoires de loi donnée.

1. Génération d'une variable aléatoire uniformément répartie sur $(0,1)$.

2. Génération d'une variable aléatoire gaussienne centrée réduite.

3. Génération d'une variable aléatoire de fonction de répartition donnée.

III - EXEMPLES NUMERIQUES.

§ 1. Exemple 1 système linéaire, coût quadratique, Bruit gaussien.

1. Le problème $P(\mu)$.

2. Solution Feedback théorique.

3. Solution open loop théorique.

4. Résolution de O_r .

4.1. temps de calcul.

4.2. commande en fonction du nombre de réalisations.

4.3. coût en fonction du nombre de réalisations.

5. Résolution de P_r et R_r .

5.1. Temps de calcul en fonction du nombre de réalisations et de la classe $\mathcal{S}_k$.

5.2. Commande initiale en fonction de l'état initial.

5.3. Coût optimal en fonction de l'état initial du nombre de réalisations de la classe $\mathcal{S}_k$.

5.4. Commandes aux instants 2 et 3 en fonction de l'état.

6. Conclusion.

§ 2. Exemple 2 : Système Production-Stock d'un produit.

1. Le problème $P(\mu)$.

2. Résultats obtenus par la programmation dynamique classique.

2.2.1. Discrétisation du bruit (7).

2.2.2. Discrétisation du bruit (7').

3. Résolution des problèmes open loop O_r^1 et O_r^2 .

2.3.1. Résultats obtenus par l'algorithme d'UZAWA appliqué au problème O_r^1 .

2.3.2. Le problème O_r^2 .

4. Résolution des problèmes relaxés R_2^1 et R_r^2 .

4.1. Algorithme d'UZAWA appliqué au problème R_r^1.

4.2. Algorithme d'UZAWA utilisé pour résoudre le problème de décomposition associé au problème R_r^1 .

4.3. Algorithme de pénalisation appliqué au problème R_r^2 .

5. Résolution du problème Feedback P_r^2 .

6. Conclusion.

§ 3. Exemple 3 : Système Production-Stock 1 ressource 5 produits.

1. Le problème $P(\mu)$

2. Résultats de l'utilisation de l'algorithme P_r en temps réel.

2.1. Ressource.

2.2. Produit 1.

2.3. Produit 2.

2.4. Produit 3.

2.5. Produit 4.

2.6. Produit 5.

§ 4. Conclusion des exemples numériques.

BIBLIOGRAPHIE.

CHAPITRE I

METHODES DE SIMULATION EN PROGRAMMATION DYNAMIQUE STOCHASTIQUE.

(Résulte d'un travail commun avec M. VIOT cf. (17)).

INTRODUCTION

On étudie des problèmes de programmation dynamique stochastique avec contraintes presque-sûre, en probabilité, ou en moyenne. La valeur Φ ou coût minimum, de tels problèmes, dépend de la mesure de probabilité μ , associée aux paramètres aléatoires (ou bruit) du système, ainsi que, des bornes α des contraintes, à savoir $\Phi(\mu,\alpha)$. On montre que, pour un choix convenable de l'ensemble des stratégies feed-back utilisées dans le contrôle du système, cette fonction Φ est semi-continue inférieurement (s.c.i.), pour la topologie de la convergence étroite sur les mesures. De plus on montre qu'elle possède une propriété de continuité, le long de certaines suites (μ_r,α_r) correctement choisies. C'est en particulier le cas, pour les mesures discrètes (notées $\mu_r(\omega)$) issues de simulations indépendantes sur le bruit du système. Nous avons appelé cette propriété : loi forte des grands nombres en programmation dynamique stochastique.

§ 1. SYSTEMES ET STRATEGIES.

Pour tout $i = 0, 1, \ldots, N$, (respectivement $i = 0, 1, \ldots, N-1$) soient n_i, (respectivement m_i et p_i), des entiers $\geqslant 1$, et f_i, des applications continues de $R^{n_i} \times R^{m_i} \times R^{p_i}$ dans $R^{n_{i+1}}$. On considère un système dont l'état est donné par l'équation de récurrence :

$$x_{i+1} = f_i(x_i, u_i, e_i) \; ; \; i = 0, 1, \ldots, N-1. \tag{1}$$

Les paramètres aléatoires du système seront les vecteurs $e = (x_o, e_o, e_1, \ldots, e_{N-1})$ de l'espace produit :

$$E = R^{n_o} \times \prod_{i=0}^{N-1} R^{p_i} .$$

Par la suite on désigne par μ une mesure de probabilité de Radon sur E.

La partie observable du système est définie de la manière suivante : on suppose que $n_i = n_i^{ob} + n_i^{inob}$ avec $n_i^{ob} \geqslant 0$; soit alors $(x_o, x_1, \ldots, x_N)$ une solution de l'équation (1) et x_i^{ob}, les n_i^{ob} premières composantes de x_i : le vecteur x_i^{ob} correspond à la partie observable du système à l'instant i. Lorsque $n_i^{ob} = 0$, le système est dit inobservable à l'instant i.

On note Σ_i , l'espace des applications continues de $R^{n_i^{ob}}$ dans R^{m_i}, muni de la topologie de la convergence compacte. Lorsque $n_i^{ob} = 0$, Σ_i s'identifie à R^{m_i}. Soit par ailleurs Σ l'espace produit

$$\prod_{i=1}^{N-1} \Sigma_i$$

(muni de la topologie produit) : un élément $S = (S_o, S_1 \ldots S_{N-1})$ de Σ sera appelé stratégie.

Remarquons que Σ est métrisable séparable et que, d'après le théorème d'Ascoli, une partie $\mathcal{S}$ de Σ est relativement compacte si et seulement si $\mathcal{S}_i = \text{proj.}_{\Sigma_i} \mathcal{S}$ est faiblement borné dans Σ_i ; $\mathcal{S}_i$ est une partie équicontinue de Σ_i.

A une stratégie $S \in \Sigma$ et à un vecteur $e=(x_o, e_o, \ldots e_{N-1})$ de E, est associée une solution de l'équation (1), notée $X(S,e) = (X_o(S,e), X_1(S,e), \ldots, X_N(S,e))$ par les relations :

$$(2) \qquad \begin{cases} X_o(S,e) = x_o \\ X_{i+1}(S,e) = f_i(X_i(S,e), S_i(X_i^{ob}(S,e)), e_i) \end{cases}$$

Une solution $X(S,e)$ est donc un élément de l'espace produit

$$F = \prod_{i=0}^{N} R^{n_i}$$

Remarques.

1. Lorsque le système est inobservable à tout instant ($n_i^{ob} = 0$, $i = 0, 1, \ldots, N-1$), les stratégies obtenues ne sont autres que les stratégies open-loop.

2. En apparence, on se limite à des stratégies markoviennes (ne dépendant que de l'observation à l'instant considéré). En fait, il suffirait d'augmenter l'espace des états de l'équation (1), en gardant en mémoire tout ou une partie des états précédents, pour obtenir des stratégies non markoviennes.

Nous allons étudier maintenant certaines propriétés de l'application $X : (S,e) \to X(S,e)$ de $\Sigma \times E$ dans F qui a été définie en (2).

Proposition 1. L'application $X : (S,e) \to X(S,e)$ est continue de $\Sigma \times E$ dans F.

Démonstration.

L'espace Σ étant métrisable (et E de dimension finie), il suffit de montrer que pour toute suite (S_r, e^r) de $\Sigma \times E$ convergeant vers (S,e), la suite $X(S^r, e^r)$ converge vers $X(S,e)$. En fait on va montrer par récurrence sur $i=0,1,\ldots,N$ que l'on a :

$$\lim_r X_i(S_r, e^r) = X_i(S,e).$$

Or cela est vrai pour $i=0$, car d'après (2) :

$$X_o(S_r, e^r) = x_o^r \quad .$$

Supposons donc la propriété vérifiée pour i, et désignons par L_i^{ob} un compact de $R^{n_i^{ob}}$ contenant la suite convergente :

$$x_i^r = X_i^{ob}(S_r, e^r)$$

Comme la suite $S_{r,i}$ converge vers S_i uniformément sur tout compact, on a :

(3) $$\forall\, \varepsilon > 0,\ \exists\, R(\varepsilon) \text{ tel que } r \geqslant R(\varepsilon) \Longrightarrow \sup_{x \in L_i^{ob}} \left| S_{r,i}(x) - S_i(x) \right| \leqslant \varepsilon \ .$$

De la majoration :

$$|S_{r,i}(x_i^r) - S_i(x_i)| \leqslant |S_{r,i}(x_i^r) - S_i(x_i^r)| + |S_i(x_i^r) - S_i(x_i)|$$

et de la condition (3), on déduit que la suite $S_{r,i}(x_i^r)$

est convergente et de limite $S_i(x_i) = S_i(X_i^{ob}(S,e))$.

La continuité est applications f_i entraînant finalement que .

$$\lim_r X_{i+1}(S_r, e^r) = X_{i+1}(S,e).$$

Corollaire. Soient $\mathcal{S}$ un compact de Σ et K un compact de E ; il existe alors un compact L de F tel que :

$$\forall (S,e) \in \mathcal{S} \times K \; ; \; X(S,e) \in L$$

On en déduit en particulier que si μ est une mesure à support compact de E et si $\mathcal{S}$ est une partie compacte de Σ, l'ensemble des vecteurs aléatoires :

$\{X(S,.) \; ; \; S \in \mathcal{S}\}$ est une partie bornée de $L^\infty(E,\mu \; ; \; F)$.

De manière plus générale, désignons par K_n une suite croissante de compacts de E telle que $\mu(\complement K_n) \leqslant \frac{1}{2^n}$ et posons :

$$\lambda_n = \max_{S \in \mathcal{S}, e \in K_n} |X(S,e)| \; .$$

Alors l'ensemble $\{X(S,.) | S \in \mathcal{S}\}$ formera une partie bornée de $L^p(E,\mu \; ; \; F)$ à condition que :

$$\sum_n \frac{\lambda_n^p}{2^n} < +\infty \; .$$

Notons $\mathcal{M}_+^1(E)$ l'ensemble des mesures de probabilité de Radon sur E, muni de la topologie de la convergence étroite : c'est un espace métrisable de type dénombrable (cf.[2] Ch.IX § 5 n° 4,p. 62).

Dans la suite un rôle essentiel sera joué par les intégrales de la forme :

$$G(S,\mu) = \int_E g(X(S,e),S)\, d\mu(e) \tag{4}$$

où

- $X(S,e)$ est la solution de (2) assciée à $S \in \Sigma$ et $e \in E$
- g une fonction s.c.i. et bornée inférieurement sur $F \times \Sigma$
- μ une mesure de $\mathcal{M}_+^1(E)$.

Proposition 2. Soit g une fonction continue sur $F \times \Sigma$ et $\mathcal{S}$ une partie de Σ. On suppose qu'il existe une fonction $\geqslant 0$, μ-intégrable, ρ telle que :

$$\forall\, S \in \mathcal{S} : |g(X(S,e),S| \leqslant \rho(e), \quad \mu\text{-p.p.} \tag{5}$$

La fonction $S \to G(S,\mu)$ est alors continue sur $\mathcal{S}$.

Démonstration. Soit S_r une suite de $\mathcal{S}$ convergeant vers un élément $S \in \mathcal{S}$. D'après la Proposition 1, on a :

$$\forall\, e \in E, \lim_{r\to\infty} g(X(S_r,e),S_r) = g(X(S,e),S)$$

De plus, la majoration (5) donne pour tout r :

$$|g(X(S_r,e),S_r)| \leqslant \rho(e) \quad , \quad \mu\text{-p.p.}$$

Le résultat découle donc du Théorème de convergence dominée de Lebesgue.

Proposition 3. Soit g une fonction s.c.i. et bornée inférieurement sur $F \times \Sigma$, et soit G la fonction définie sur $\Sigma \times \mathcal{M}_+^1(E)$ par (4).

Alors G est s.c.i. et bornée inférieurement. De plus G est continue bornée si g est continue bornée.

Démonstration. Supposons d'abord g continue bornée et posons

$$\tilde{g}(e,S) = g(X(S,e),S)$$

La fonction $\tilde{g}$ définie sur $E \times \Sigma$ est également continue bornée :

$$\forall\, e \in E\,,\ \forall\, S \in \Sigma\ :\ |\tilde{g}(e,S)| \leqslant M\ .$$

Soit alors (S_r,μ_r) une suite de $\Sigma \times \mathcal{M}_+^1(E)$ convergeant vers (S,μ) ; on a :

$$|G(S_r,\mu_r)-G(S,\mu)| \leqslant \left|\int \tilde{g}(e,S)d\mu_r(e)-d\mu(e)\right|+\left|\int \tilde{g}(e,S_r)-\tilde{g}(e,S)d\mu_r(e)\right|$$

Le premier terme du second membre de l'inégalité tend vers 0 par définition même de la convergence étroite.

Considérons maintenant un compact K de E tel que $\mu(E-K) \leqslant \varepsilon$ et soit L un ouvert relativement compact contenant K. Il existe $R_1(\varepsilon)$ tel que :

$$r \geqslant R_1(\varepsilon) \Rightarrow \mu_r(E-L) \leqslant \varepsilon$$

Par ailleurs, comme $\tilde{g}$ est continue, il existe $R_2(\varepsilon)$ tel que :

$$r \geqslant R_2(\varepsilon) \Rightarrow \sup_{e \in L} |\tilde{g}(e,S_r)-\tilde{g}(e,S)| \leqslant \varepsilon\ .$$

Donc pour $r \geqslant \max(R_1(\varepsilon), R_2(\varepsilon))$ on a

$$\left|\int \tilde{g}(e,S_r)-\tilde{g}(e,S)\ d\mu_r(e)\right| \leqslant 2M\ \mu_r(E-L) + \varepsilon\ \mu_r(L)$$

$$\leqslant (2M+1)\varepsilon\ .$$

D'où la continuité de G lorsque g est continue bornée.

Supposons maintenant g s.c.i. et bornée inférieurement. Les espaces F et Σ étant métrisables, la fonction g est l'enveloppe supérieure d'une suite croissante (g_r) de fonctions continues bornées. Et le résultat découle de la relation :

$$G(S,\mu) = \sup_r \int_E g_r(X(S,e),S)\ d\mu(e).$$

Exemples.

1° Soit $\mathcal{O}$ un ouvert de F, alors la fonction :

$$(S,\mu) \rightarrow \mu\{e : X(S,e) \in \mathcal{O}\}$$

est s.c.i. sur $\Sigma \times \mathcal{M}_+^1(E)$.
Il suffit d'appliquer la Proposition 3 à g = fonction caractéristique de l'ouvert $\mathcal{O}$.

2° Soit B un fermé de F et α un scalaire, $0 \leqslant \alpha \leqslant 1$.
La fonction :

$$(S,\mu) \rightarrow \alpha - \mu\{e : X(S,e) \in B\}$$

est s.c.i. sur $\Sigma \times \mathcal{M}_+^1(E)$.
(Appliquer la Proposition 3 à $g = \alpha$ - f.caractéristique (B)).

3° Soient p_i $(i = 1,\ldots,N)$ des fonctions s.c.i. et bornées inférieurement sur $R^{n_i} \times R^{m_{i-1}}$; la fonction :

$$(S,\mu) \to \sum_{i=1}^{N} \int_E p_i[X_i(S,e), S_{i-1}(X_{i-1}^{ob}(S,e))]\, d\mu(e)$$

est également s.c.i. sur $\Sigma \times \mathcal{M}_+^1(E)$.

En conséquence les intégrales du type (4) englobent les principales formes de contraintes presque-sûre en probabilité ou en moyenne, de même que les principales formes de critères que l'on rencontre en programmation dynamique stochastique.

Soient donc $g_0, g_1, \ldots, g_K$, K+1 fonctions s.c.i. et bornées inférieurement sur $F \times \Sigma$, $\alpha = (\alpha^1 \ldots \alpha^K)$ un point de R^K, μ une mesure de $\mathcal{M}_+^1(E)$ et $\mathfrak{s}$ une partie de Σ ; un problème de programmation dynamique stochastique s'écrit sous la forme :

$$(6) \qquad P(\mu,\alpha) \begin{cases} \Phi(\mu,\alpha) = \inf\limits_{S \in \mathfrak{s}} G_0(S,\mu) \\ X_{i+1}(S,e) = f_i(X_i(S,e),\ S_i(X_i^{ob}(S,e)), e_i) \\ G_k(S,\mu) \leqslant \alpha_k \qquad k = 1,\ldots,K \end{cases}$$

En particulier lorsque $\mathfrak{s}$ est une partie compact de Σ et que l'ensemble des stratégies admissibles est non vide, on est assuré de l'existence d'une stratégie optimale pour $P(\mu,\alpha)$ (cf. Proposition 3).

Orientation générale.

Lorsque la mesure μ sur E est discrète : $\mu = \sum_{j=1}^{r} p_j \delta_{e^j}$, le programme (6) s'écrit simplement :

$$(6') \qquad P(\mu,\alpha)\begin{cases} \Phi(\mu,\alpha) = \inf_{S\in\mathcal{S}} \sum_{j=1}^{r} p_j\, g_o(X(S,e^j),S) \\ X_{i+1}(S,e^j) = f_i(X_i(S,e^j),S_i(X_i^{ob}(S,e^j),e_i^j) \\ \sum_{j=1}^{r} p_j\, g_k(X(S,e^j),S) \leqslant \alpha_k \quad , \ k = 1,\ldots, K \end{cases}$$

On peut donc espérer le résoudre (au moins dans certains cas), par des méthodes purement déterministes.

Par ailleurs, on sait que l'ensemble des mesures discrètes est dense dans $\mathcal{M}_+^1(E)$. Si donc la valeur Φ était une fonction continue sur $\mathcal{M}_+^1(E)$ on aurait ainsi une méthode d'approximation simple des problèmes (6) par discrétisation de la mesure de probabilité des bruits du système.

En fait on ne peut espérer mieux qu'une semi-continuité inférieure globale pour la fonction Φ (Théorème 1). Le travail principal consistera donc à trouver des conditions suffisantes de continuité de Φ le long de certaines suites (Proposition 7) et à montrer que ces conditions suffisantes sont réalisées dans le cas particulièrement intéressant de la simulation indépendante des paramètres aléatoires du système (§ 3).

§ 2. QUELQUES PROPRIETES DE LA VALEUR Φ .

Dans ce paragraphe $\mathcal{S}$ désignera une partie compact de Σ .

Soit $\Gamma(\mu,\alpha)$, l'ensemble des stratégies admissibles du programme $P(\mu,\alpha)$:

$$\Gamma(\mu,\alpha) = \{S \in \mathcal{S} : G_k(S,\mu) \leqslant \alpha_k,\ k = 1,\dots K\} \tag{7}$$

Proposition 4. La multi-application Γ est semi-continue supérieurement de $\mathcal{M}_+^1(E) \times R^k$ dans $\mathcal{S}$.

Démonstration. Les fonctions G_k étant s.c.i. et $\mathcal{S}$ compact, les $\Gamma(\mu,\alpha)$ sont des compacts.

Soit (S_r,μ_r,α_r) une suite de points du graphe de Γ, convergeant vers (S,μ,α) ; on a donc pour tout $k = 1 \dots K$:

$$G_k(S,\mu) \leqslant \lim_r \inf G_k(S_r,\mu_r) \leqslant \lim_r \inf \alpha_r = \alpha$$

D'où $S \in \Gamma(\mu,\alpha)$. En conséquence Γ est de graphe fermé, ce qui entraîne, $\mathcal{S}$ étant compact, la s.c.s. de Γ([1] ch. VI cor. Théor. 7).

Corollaire. Soit (μ_r,α_r) une suite de $\mathcal{M}_+^1(E) \times \mathbb{R}^K$ convergeant vers (μ,α). Alors :

$$\overline{\lim_r \inf \Gamma(\mu_r,\alpha_r)} \subset \overline{\lim_r \sup \Gamma(\mu_r,\alpha_r)} \subset \Gamma(\mu,\alpha) \tag{8}$$

Démonstration. Il suffit de montrer que :

$$\lim_r \sup \Gamma(\mu_r,\alpha_r) \subset \Gamma(\mu,\alpha)$$

Dire que $S \in \lim_r \sup \Gamma(\mu_r,\alpha_r)$ entraîne qu'il existe une sous-suite (r_j) telle que $S \in \Gamma(\mu_{r_j},\alpha_{r_j})$ pour tout r_j. D'où pour tout $k=1,\dots,K$:

$$G_k(S,\mu) \leqslant \lim_{r_j} \inf G_k(S,\mu_{r_j}) \leqslant \lim_{r_j} \inf \alpha_{r_j} = \alpha$$

Et on a bien $S \in \Gamma(\mu,\alpha)$.

Théorème 1. La valeur $\Phi(\mu,\alpha)$ de $P(\mu,\alpha)$ est une fonction semi-continue inférieurement et bornée inférieurement sur $\mathcal{M}_+^1(E) \times \mathbb{R}^K$.

Démonstration.

1°. Posons $\Lambda = \{(\mu,\alpha) \in \mathcal{M}_+^1(E) \times \mathbb{R}^K \mid \Gamma(\mu,\alpha) \neq \emptyset\}$ et montrons que Λ est fermé.

Soit $(\mu_r,\alpha_r) \in \Lambda$ convergeant vers (μ,α) et pour tout r, soit S_r un élément de $\Gamma(\mu_r,\alpha_r)$.

Comme $\mathcal{S}$ est compact on peut toujours supposer la suite (S_r) convergente vers un $S \in \mathcal{S}$. La s.c.s. de Γ entraîne alors que $S \in \Gamma(\mu,\alpha)$. D'où $(\mu,\alpha) \in \Lambda$.

2° La restriction de Γ à Λ étant s.c.s. et non vide, la restriction de Φ à Λ est s.c.i. d'après [1], (chap. VI, § 3, Th. 2).

Et comme en dehors du fermé Λ, Φ est identique à $+\infty$, la la fonction Φ est donc partout s.c.i.

Proposition 5. Soit (μ_r,α_r) une suite de $\mathcal{M}_+^1(E) \times \mathbb{R}^K$ convergeant vers (μ,α) et soit $\tilde{S}_r$ une solution optimale du problème $P(\mu_r,\alpha_r)$. Supposons que

$$\liminf_r \Phi(\mu_r,\alpha_r) = \Phi(\mu,\alpha) \tag{9}$$

il existe alors une sous-suite de $(\tilde{S}_r)$ convergeant vers une solution optimale de $P(\mu,\alpha)$.*

---

* On rappelle que μ_r converge vers μ pour la topologie de la convergence étroite et que la sous-suite de $(\tilde{S}_r)$ convergera au sens de la topologie de la convergence compacte sur Σ.

Démonstration.

Posons : $\beta_r = \Phi(\mu_r, \alpha_r) = G_o(\tilde{S}_r, \mu_r)$

et $\beta = \lim_r \inf \Phi(\mu_r, \alpha_r) = \Phi(\mu, \alpha)$.

Soit $(\beta_{r_j'})$ une sous-suite de (β_r) telle que : $\lim_{r_j'} \beta_{r_j'} = \beta$.

De la suite $(\tilde{S}_{r_j'})$ on peut extraire une sous-suite $(\tilde{S}_{r_j})$ convergeant vers un certain $\tilde{S} \in \Gamma(\mu, \alpha)$, d'après la s.c.s. de Γ. Enfin la s.c.i. du critère G_o entraîne :

$$G_o(\tilde{S}, \mu) \leqslant \lim_r \inf G_o(\tilde{S}_{r_j}, \mu_{r_j}) = \lim_{r_j} \inf \Phi(\mu_{r_j}, \alpha_{r_j}) = \Phi(\mu, \alpha).$$

Donc $\tilde{S}$ est une solution optimale de $P(\mu, \alpha)$.

Corollaire. Si on remplace l'hypothèse (9) par :

(10) $$\lim_r \Phi(\mu_r, \alpha_r) = \Phi(\mu, \alpha)$$

alors toute sous-suite convergente de $(\tilde{S}_r)$ converge vers une solution optimale de $P(\mu, \alpha)$.

Démonstration.

Si $(\tilde{S}_{r_j})$ converge vers $\tilde{S}$, on a $\tilde{S} \in \Gamma(\mu, \alpha)$ et de plus :

$$G_o(\tilde{S}, \mu) \leqslant \lim_{r_j} \inf \Phi(\mu_{r_j}, \alpha_{r_j}) = \Phi(\mu, \alpha)$$

Proposition 6. Soit (μ_r, α_r) une suite convergeant vers (μ, α) et supposons que :

a) l'application $S \to G_o(S, \mu)$ est continue sur $\mathcal{S}$

b) il existe une partie dénombrable : $\Gamma^d \subset \lim_r \sup \Gamma(\mu_r, \alpha_r)$ dense dans $\Gamma(\mu, \alpha)$ et telle que :

$$\forall\ S \in \Gamma^d \quad , \quad \lim_r G_o(S,\mu_r) = G_o(S,\mu)$$

Alors on a : $\liminf_r \Phi(\mu_r,\alpha_r) = \Phi(\mu,\alpha)$ et si $(\tilde{S}_r)$ est optimale pour $P(\mu_r,\alpha_r)$ ^{r} il existe une sous-suite de $(\tilde{S}_r)$ convergeant vers une solution optimale de $P(\mu,\alpha)$.

Démonstration.

Pour $S \in \Gamma^d$, il existe (r_j) telle que $S \in \Gamma(\mu_{r_j},\alpha_{r_j})$ et donc :

$$\liminf_r \Phi(\mu_r,\alpha_r) \leqslant \lim_{r_j} G_o(S,\mu_{r_j}) = G_o(S,\mu)$$

La continuité de $G_o(\ .,\mu)$ et la condition b) entraînent alors que :

$$\liminf_r \Phi(\mu_r,\alpha_r) \leqslant G_o(S,\mu) \quad , \quad \forall\ S \in \Gamma(\mu,\alpha)$$

D'où l'inégalité : $\liminf_r \Phi(\mu_r,\alpha_r) \leqslant \Phi(\mu,\alpha)$
L'inégalité inverse découle de la s.c.i. de la fonction Φ (Théor. 1) et le reste de la Propriété 5.

Proposition 7 . Soit (μ_r,α_r) une suite convergeant vers (μ,α) et supposons que :

a) l'application $S \to G_o(S,\mu)$ est continue sur $\mathcal{S}$

b) il existe une partie dénombrable $\Gamma^d \subset \liminf_r \Gamma(\mu_r,\alpha_r)$ et dense dans $\Gamma(\mu,\alpha)$, telle que

$$\forall\ S \in \Gamma^d \quad , \quad \lim_r G_o(S,\mu_r) = G_o(S,\mu)$$

Alors on a : $\lim_r \Phi(\mu_r,\alpha_r) = \Phi(\mu,\alpha)$ et toute sous-suite convergente de $(\tilde{S}_r)$ ^{r}, converge vers une solution optimale de $P(\mu,\alpha)$.

Démonstration.

Soit (β_{r_j}) une sous-suite convergente de la suite $\beta_r = \Phi(\mu_r, \alpha_r)$ et soit $S \in \Gamma^d$. Comme $S \in \liminf_r \Gamma(\mu_r, \alpha_r)$, à partir d'un certain rang $r_j \geqslant r_o$, on aura $S \in \Gamma(\mu_{r_j}, \alpha_{r_j})$. D'où :

$$\beta = \lim_{r_j} \Phi(\mu_{r_j}, \alpha_{r_j}) \leqslant \lim_{r_j} G_o(S, \mu_{r_j}) = G_o(S, \mu).$$

Par continuité de $G_o(\ .\ , \mu)$ on obtient encore :

$$\beta \leqslant G_o(S, \mu) \quad , \quad \forall\ S \in \Gamma(\mu, \alpha) .$$

Donc tout point d'accumulation de la suite (β_r) est inférieur ou égal à $\Phi(\mu, \alpha)$; mais comme on a toujours :

$$\liminf_r \beta_r \geqslant \Phi(\mu, \alpha) \quad \text{(Théorème 1)},$$

cela n'est possible que si $\Phi(\mu, \alpha) = \lim_r \Phi(\mu_r, \alpha_r)$.

Le reste découle alors du corollaire, Proposition 5.

---

§ 3. LOI FORTE DES GRANDS NOMBRES EN PROGRAMMATION DYNAMIQUE STOCHASTIQUE.

.1. Rappels sur la loi forte des grands nombres.

Soit $(\Omega, \mathcal{F}, P)$ un espace probabilisé et (η_j) $j \geqslant 1$, une suite de variables aléatoires indépendantes équidistribuées, définies sur cet espace. On suppose que la loi commune des η_j admet un moment du 1er ordre a, et un écart type σ fini.

Posons :

$\xi_r = \sum_{j=1}^{r} (\eta_j - a)$. La martingale ξ_r possède alors les propriétés suivantes :

Proposition 8 : Pour tout réel $\varepsilon > 0$, on a :

$$\lim_r \frac{1}{r^{\frac{1}{2}+\varepsilon}} \sum_{j=1}^{r} (\eta_j - a) = 0 \quad \text{(p.s.)} . \tag{11}$$

Démonstration.

Soit $u_r = r^{\frac{1}{2}+\varepsilon}$. La série $\sum_r u_r^{-2}\sigma$, étant convergente, le résultat découle de [3], Proposition IV-6-1.

Remarquons que la condition (11) est équivalente à :

$$\forall\, \lambda > 0 \,,\; P\left(\lim_r \inf \left\{\omega : \left|\frac{1}{r}\, \xi_r\right| \leqslant \lambda\, r^{-\frac{1}{2}+\varepsilon}\right\}\right) = 1 \tag{11'}$$

Proposition 9. Supposons que la loi commune des η_j est à support compact. Alors :

(12) $$P(\limsup_r \xi_r = +\infty) = P(\liminf_r \xi_r = -\infty) = 1$$

(pour la démonstration voir [3] Prop. IV.6.3).

Une conséquence de cette dernière proposition est que, si on désigne par $\bar{\eta}_r = \frac{1}{r} \sum_{j=1}^{r} \eta_j$, la moyenne arithmétique de la suite (η_j), la suite $\bar{\eta}_r$ converge presque sûrement vers a en oscillant une infinité dénombrable de fois autour de cette valeur :

$$P(\limsup_r \{\bar{\eta}_r \leqslant a\}) \geqslant P(\liminf_r \xi_r < 0) = 1$$

$$P(\limsup_r \{\bar{\eta}_r \geqslant a\}) \geqslant P(\limsup_r \xi_r > 0) = 1$$

2. Application à la simulation indépendante.

Dans toute la suite de cette partie, on désignera par $P(\mu, \alpha)$ un problème de programmation dynamique stochastique vérifiant les hypothèses supplémentaires :

H_1 - le critère $G_0(S,\mu)$ est continu sur $\mathcal{S}$ (μ fixé).

Une condition suffisante pour que H_1 soit vérifiée est donnée par la Proposition 2.

H_2 - pour tout $k = 1, \ldots, K$ et tout $S \in \mathcal{S}$:

$$\int_E g_k^2 (X(S,e),S)\, d\mu(e) < +\infty$$

Par ailleurs on notera $(\Omega, \mathcal{F}, P)$ l'espace produit $(E^{\mathbb{N}}, \bigotimes_{n\in\mathbb{N}} \mathcal{B}, \bigotimes_{n\in\mathbb{N}} \mu)$: c'est donc l'espace des simulations (ou tirages) indépendantes des paramètres aléatoires e du système.

A tout $\omega = (e_r)_{r \geqslant 1} \in \Omega$, on associe une suite de mesures discrètes $(\mu_r(\omega))$ sur E, en posant :

$$\mu_r(\omega) = \frac{1}{r} \sum_{j=1}^{r} \delta_{e^j} \tag{13}$$

D'après le théorème de Glivenko-Cantelli ([4] Ch. 10 § 10), la suite $\mu_r(\omega)$ converge étroitement vers μ, pour presque tout $\omega \in \Omega$.

Par ailleurs remarquons que $G_k(S, \mu_r(\omega))$ s'écrit :

$$G_k(S, \mu_r(\omega)) = \frac{1}{r} \sum_{j=1}^{r} g_k(S, e^j) \; ; \; k = 0, 1, \ldots, K \; . \tag{14}$$

C'est donc la moyenne arithmétique de la suite de variables aléatoires indépendantes équidistribuées $\eta_k^j(\omega) = g_k(S, e^j)$.

On étudie maintenant l'approximation de $P(\mu, \alpha)$ par des problèmes $P(\mu_r(\omega), \alpha_r)$.

Théorème 2 . Supposons les hypothèses H_1 , H_2, vérifiées et soit α_r une suite de R^K convergeant vers α de telle façon qu'il existe $\varepsilon > 0$ et $\lambda > 0$ tels que :

$$\alpha_k^r - \alpha_k \geqslant \lambda \, r^{-\frac{1}{2}+\varepsilon} \; ; \quad k = 1, \ldots, K.$$

Soit $\tilde{S}_r(\omega)$ une solution optimale du problème $P(\mu_r(\omega), \alpha^r)$, lorsque $\Gamma(\mu_r(\omega), \alpha^r)$ est non vide, ou dans le cas contraire un élément quelconque de $\mathcal{S}$.
Alors sauf sur un P négligeable de Ω, on a :

a) $$\Phi(\mu, \alpha) = \lim_r \Phi(\mu_r(\omega), \alpha^r)$$

b) toute sous-suite convergente de $\tilde{S}_r(\omega)$, converge vers une solution optimale de $P(\mu, \alpha)$.

Démonstration.

Il s'agit de montrer que pour presque tout $\omega \in \Omega$, les conditions a) et b) de la Proposition 7 sont vérifiées.

La condition a) découle de l'hypothèse H_1.

Posons maintenant ;

$$\varepsilon_k^r = \alpha_k^r - \alpha_k$$

$$A_k^r(S) = \{\omega \in \Omega : | G_k(S,\mu_r(\omega)) - G_k(S,\mu)| \leqslant \varepsilon_k^r\}$$

D'après (14) et (11'), la moyenne arithmétique $G_k(S,\mu_r(\omega))$ va converger p.s. vers $G_k(S,\mu)$ avec une vitesse supérieure à $r^{-\frac{1}{2}+\varepsilon} \leqslant \varepsilon_k^r$.

Donc pour tout S, et tout k :

$$P(\liminf_r A_k^r(S)) = 1 \tag{15}$$

Soit par ailleurs Γ^d une partie dénombrable dense dans $\Gamma(\mu,\alpha)$. On a les inclusions :

$$\Omega_1 = \{\omega : \Gamma^d \subset \liminf_r \Gamma(\mu_r(\omega),\alpha^r)\}.$$

$$= \bigcap_{S\in\Gamma^d} \liminf_r \{\omega : S \in \Gamma(\mu_r(\omega),\alpha^r)\}$$

$$\supset \bigcap_{S\in\Gamma^d} \liminf_r \left(\bigcap_{k=1}^K A_k^r(S)\right)$$

$$= \bigcap_{S\in\Gamma^d} \bigcap_{k=1}^K \liminf_r A_k^r(S)$$

En conséquence, d'après (15) :

$$P(\Omega_1) = 1$$

Désignons maintenant par Ω_2 l'ensemble des ω tels que $\mu_r(\omega)$ converge étroitement vers μ, et par Ω_3 :

$$\Omega_3 = \{\omega : G_o(S,\mu) = \lim_r G_o(S,\mu_r(\omega)) \ , \ \forall\ S \in \Gamma^d\} \ .$$

On sait que $P(\Omega_2) = 1$ (Thèorème de Glivenko-Cantelli) et de même $P(\Omega_3) = 1$ d'après le loi forte des grands nombres et le fait que Γ^d est dénombrable.

Posons alors $\Omega_o = \Omega_1 \cap \Omega_2 \cap \Omega_3$. Dans ces conditions, on a montré que $P(\Omega_o) = 1$ et que pour tout $\omega \in \Omega_o$, la suite $(\mu_r(\omega),\alpha^r)$ converge vers (μ,α) en vérifiant :

$$\begin{cases} - \ \Gamma^d \subset \lim_r \inf \Gamma\,(\mu_r(\omega),\alpha^r) \\ - \ \Gamma^d \text{ dense dans } \Gamma(\mu,\alpha) \\ - \ \forall\ S \in \Gamma^d,\ \lim_r G_o(S,\mu_r(\omega)) = G_o(S,\mu) \end{cases}$$

c.q.f.d.

REMARQUE. Dans la pratique, ce résultat a un inconvénient : celui de perturber les bornes des contraintes. En particulier, si on part d'une contrainte presque-sûre pour $P(\mu,\alpha)$:

$$\mu\ \{\ e : X(S,e) \in B\ \} \ = \ 1$$

cette contrainte est transformée en contrainte en probabilité pour $P(\mu_r(\omega),\ \alpha^r)$:

$$\frac{1}{r} \text{ card } \{1 \leqslant j \leqslant r : X(S,e^j) \in B\} \ \geqslant 1 - \varepsilon_r$$

avec $$\varepsilon_r > r^{-\frac{1}{2}} + \varepsilon .$$

3. Cas de contraintes presque-sûres.

On considère ici un problème $P(\mu)$ vérifiant l'hypothèse H_1 et dont toutes les contraintes sont du type presque-sûre. Dans ce cas, on peut toujours se ramener à la donnée d'un sous-ensemble B de F tel que l'on ait l'unique contrainte :

$$\mu \{ e : X(S,e) \in B \} = 1.$$

Dans ces conditions, le programme s'écrit :

$$P(\mu) \begin{cases} \Phi(\mu) = \min\limits_{S \in \mathcal{S}} G_o(S,\mu) \\ X_{i+1}(S,e) = f_i(X_i(S,e), S_i(X_i^{ob}(S,e)), e) \\ \mu \{ e : X(S,e) \in B \} = 1 . \end{cases}$$

Et les stratégies admissibles sont données par :

$$\Gamma(\mu) = \{ S \in \mathcal{S} : \mu(X(S,e) \in B) = 1 \}$$

Remarquons que les problèmes simulés : $P(\mu_r(\omega))$, $\omega = (e_j)_{j \geq 1}$, se mettent sous la forme simple :

$$P(\mu_r(\omega)) \begin{cases} \Phi(\mu_r(\omega)) = \min\limits_{S \in \mathcal{S}} \frac{1}{r} \sum\limits_{j=1}^{r} g_o(S,e^j) \\ X_{i+1}(S,e^j) = f_i(X_i(S,e^j), S_i(X_i^{ob}(S,e^j)), e^j) \\ X(S,e^j) \in B \qquad 1 \leq j \leq r. \end{cases}$$

Théorème 3. Supposons que H_1 est vérifié et que B soit fermé dans F.

Soit $\tilde{S}_r(\omega)$ une solution optimale de $P(\mu_r(\omega))$, lorsque $\Gamma(\mu_r(\omega))$ est non vide ou, dans le cas contraire, un élément quelconque de $\mathcal{S}$.

Alors, sauf sur un P-négligeable de Ω :

a) $\Phi(\mu) = \lim_r \Phi(\mu_r(\omega))$

b) toute sous-suite convergente de $\tilde{S}_r(\omega)$ converge vers une solution optimale de $P(\mu)$.

Démonstration.

On applique encore la Proposition 7. La condition a) est obtenue par l'hypothèse H_1.

Posons maintenant pour $S \in \mathcal{S}$:

$$B_S = \{ e : X(S,e) \in B \}$$

$$A_r(S) = \{ \omega : \mu_r(\omega)(B_S) \geqslant \mu(B_S) \} .$$

Remarquons que :

$$\mu_r(\omega)(B_S) = \frac{1}{r} \sum_{j=1}^{r} \chi_S(e_j)$$

où χ_S est la fonction caractéristique de B_S.

L'application de la Proposition 9, nous dit que la fréquence empirique $\mu_r(\omega)(B_S)$ va converger vers $\mu(B_S)$ en oscillant une infinité dénombrable de fois autour de cette valeur. Donc pour tout S :

$$P(\limsup_r A_r(S)) = 1 \qquad (16)$$

Soit maintenant Γ^d une partie dénombrable dense de $\Gamma(\mu)$. En utilisant la décroissance des ensembles $\Gamma(\mu_r(\omega))$, on vérifie les inclusions :

$$\Omega_1 = \{\omega : \Gamma^d \subset \lim_r \inf \Gamma(\mu_r(\omega))\}$$

$$= \bigcap_{S\in\Gamma^d} \lim_r \inf \{\omega : S \in \Gamma(\mu_r(\omega))\}$$

$$= \bigcap_{S\in\Gamma^d} \lim_r \sup \{\omega : S \in \Gamma(\mu_r(\omega))\}$$

$$\supset \bigcap_{S\in\Gamma^d} \lim_r \sup A_r(S).$$

Donc, d'après (16), on a :

$$P(\Omega_1) = 1$$

Et comme dans le Théorème 2, on pose :

$$\Omega_2 = \{\omega \in \Omega : \mu_r(\omega) \text{ converge étroitement vers } \mu\}$$

$$\Omega_3 = \{\omega \in \Omega : G_o(S,\mu) = \lim_r G_o(S,\mu_r(\omega)), \quad \forall\, S \in \Gamma^d\}$$

$$\Omega_o = \Omega_1 \cap \Omega_2 \cap \Omega_3 .$$

On a encore $P(\Omega_o) = 1$ et la condition b) de la Proposition 7 se trouve ainsi vérifiée.

Remarque. Soit T une transformation sur E laissant la mesure μ invariante. Le problème $P(\mu)$ est alors équivalent à minimiser sur $\mathcal{S}$, le nouveau critère :

$$\tilde{G}(S,\mu) = \frac{1}{2}\{ \int_E g_o(X(S,e),S)\, d\mu(e) + \int_E g_o(X(S,Te),S)\, d\mu(e)\}$$

sous la nouvelle contrainte :

$$\mu\{ e : X(S,e) \in B \} = \mu\{e : X(S,Te) \in B \} = 1.$$

Pratiquement, cela revient à ajouter l'état supplémentaire X(S,Te) au système initial. Le Théorème 3 s'applique donc encore à ce nouveau problème.

CHAPITRE II

ALGORITHMES DE RESOLUTION DES PROBLEMES $P(\mu_r)$

Dans ce chapitre nous nous proposons de résoudre numériquement les problèmes $P(\mu_r)$ où μ_r est la loi discrète construite sur des simulations indépendantes du bruit du système :

$$\mu_r(\omega) = \frac{1}{r} \sum_{j=1}^{r} \delta_{e^j} \qquad \omega = (e^j \quad j \in \mathbb{N})$$

La résolution des problèmes $P(\mu_r, \alpha_r)$ est plus difficile du fait des contraintes du type :

$$\mu_r\{ e : X(S,e) \in B\} \geqslant \alpha_r$$

Elles reviennent en effet à imposer la proportion minimale de trajectoires devant appartenir au domaine B.

Nous ne ferons que quelques remarques sur ce sujet.

Une fois réalisées les simulations du bruit, on est ramené à un problème de contrôle purement déterministe que nous résoudrons essentiellement par des algorithmes de gradient.

Nous rappellerons brièvement les méthodes qui nous ont permis de tester les résultats :

- La programmation dynamique classique,
- La méthode de KALMAN.

Nous donnerons enfin les moyens de générer numériquement les bruits.

§ 1 - LES PROBLEMES P_r, R_r, O_r .

On reformule dans ce paragraphe le programme $P(\mu_r)$.

Choisissant les fonctions k-lipschitziennes comme ensemble $\mathcal{S}_k$ des stratégies sur lequel on minimisera, on obtient le problème P_r .

Une approximation plus maniable numériquement sera noté R_r .

Le cas particulier k=0 résoudra le problème Open loop. Il sera noté O_r

1. La classe $\mathcal{S}_k$ de stratégies.

$\mathcal{S}_k$ sera l'ensemble des fonctions k-lipschitziennes

$$\mathcal{S}_k = \{ (S_i, i=0,\ldots N-1) \mid S_i : \mathbb{R}^{n_i} \to C_i \subset \mathbb{R}^{m_i} \; ; \; S_i(0) \text{ borné}$$

$$\forall\, x,x' \quad |S_i(x) - S_i(x')| \leq k|x-x'| \; \} \; (1)$$

Cet ensemble est compact d'après le théorème d'ASCOLI. Les résultats du chapitre I s'appliqueront donc.

Nous noterons dans la suite :

$$u_i = S_i(x).$$

Le cas particulier $\mathcal{S}_o$ donne les stratégies open loop

$$\mathcal{S}_o = \{ (u_i, i=0,\ldots N-1) \quad u_i \in C_i \subset \mathbb{R}^{m_i} \}$$

$\omega = (e^j \; j \geq 1)$ étant une suite de réalisations du bruit du système, notons x^j (resp u^j) la j^{eme} réalisation de l'état $x = (x_o, \ldots x_N)$ (resp de la commande $u = (u_o, \ldots u_{N-1})$).

2. Le problème P_r .

Ce sera le problème $P(\mu_r(\omega))$ associé à $\mathcal{S}_k$

---

(1) $|.|$ désignera la norme euclidienne, . le produit scalaire correspondant

- Son équation d'évolution

$$(1)\qquad \begin{cases} x_i^j = f_i(x_i^j, u_i^j, e_i^j) & i=0,\dots N-1 \\ & j = 1,\dots r \\ x_o \text{ donné} \end{cases}$$

- Ses contraintes

$$(2)\qquad p_i(x_i^j, u_i^j) \leqslant 0 \qquad \left.\begin{matrix} i=0,\dots N-1 \\ \\ j=1,\dots r \end{matrix}\right\}$$

$$(3)\qquad q_i(u_i^j) \leqslant 0$$

- Les contraintes "Feedback" provenant la classe de stratégies sur laquelle on minimise

$$(4)\qquad |u_i^j - u_i^{j'}| \leqslant k|x_i^j - x_i^{j'}| \qquad \begin{matrix} i=0,\dots N-1 \\ j\neq j' \quad j=1,\dots r \\ \qquad\quad j'=1,\dots r \end{matrix}$$

le coût associé à la commande $U = (u_i^j, i=0,\dots N-1,\ j=1,\dots r)$

$$(5)\qquad G_u^r = \frac{1}{r} \sum_{j=1}^{r} \left(\sum_{i=0}^{N-1} L_i(x_i^j, u_i^j) + M(x_N^j) \right)$$

Le problème P_r sera de minimiser le coût (5) sous les contraintes (1)(2)(3)(4). On l'appellera l'approximation Feedback du problème $P(\mu)$.

Remarque

La contrainte (4) n'est pas convexe en général. Supposons f_i $i=0,\dots N-1$ linéaire l'application $u \longrightarrow X_u$ est alors linéaire.

Notons

$$U_i^- = (u_k^j,\ k< i\ ,\ j=1,\dots r)$$

$$U_i^+ = (u_k^j,\ k\geqslant i\ ,\ j=1,\dots r)$$

alors (4) s'écrit

$$\varphi(U_i^+) \leqslant \psi(U_i^-)$$

avec φ et ψ convexes

$$\varphi : U_i^+ \longrightarrow |u_i^j - u_i^{j'}|$$

$$\psi : U_i^- \longrightarrow k|x_i^j - x_i^{j'}|$$

$\mathcal{U} = \{ u : \varphi(U_i^+) \leqslant \psi(U_i^-)\}$ n'est donc pas convexe. FACK et SOLAND (5) donne un algorithme résolvant des problèmes de minimisation où :

- Le critère est la somme de fonctions convexes et concaves
- Les contraintes sont des sommes de fonctions convexes et concaves.

Le problème P_r entre dans ce cadre si f_i linéaire ; p_i, L_i, M convexes. Par cet algorithme il est donc possible d'obtenir le minimum global . Cependant étant donné la qualité de l'approximation P_r du problème $P(\mu)$ il nous a semblé préférable d'utiliser des algorithmes de gradient. Ces derniers donneront un minimum local mais seront beaucoup plus rapides et faciles à mettre en oeuvre.

3. Le problème R_r .

Il consiste à remplacer dans P_r la contrainte (4) par :

(4') $$u_o^j = u_o \quad \forall\ j$$

Remarque 1

Pour k grand, cette approximation revient pratiquement à supposer :
En dehors de l'état initial les états correspondant à deux réalisations différentes ne sont jamais égaux. Cette supposition est souvent vérifiée lorsqu'on travaille avec peu de réalisations.

Remarque 2

L'intérêt de cette approximation est de remplacer la contrainte (4) non con-

vexe par la contrainte (4') convexe.

En remplaçant (4') dans l'équation d'évolution (1) on obtient

$$(1') \qquad \left\{ \begin{array}{l} x_1^j = f_o(x_o, u_o, e_i^j) \\ x_i^j = f_i(x_i^j, u_i^j, e_i^j) \quad i=1,\ldots,N-1 \\ x_o \quad \text{donné} \end{array} \right\} \quad j=1,\ldots r$$

Le problème R_r consistera à minimiser le coût (5) sous les contraintes (1')(2)(3). Nous l'appellerons approximation relaxée de $P(\mu)$.

4. Le problème O_r .

Ce sera l'approximation du problème open loop correspondant à $P(\mu)$. On l'obtient en faisant $k=0$ dans (4).

$$(4'') \qquad u_i^j = u_i \qquad i=0,\ldots N-1 \ ; \ j=1,\ldots r$$

Remarque 1

(4") peut être considérée comme une approximation de (4). Celle où l'on suppose a priori que les états de deux réalisations différentes sont toujours égaux.

(4") peut donc être considérée comme l'approximation de (4) "opposée" à celle faite dans le problème relaxé.

Selon que les réalisations de trajectoires ont tendance à s'agglomérer ou à se disperser on aura donc intérêt à remplacer (4) par (4") ou (4').

Remarque 2

(4") est également une contrainte convexe. En remplaçant (4") dans l'équation d'évolution (1) on obtient

$$(1'') \qquad \begin{cases} x_i^j = f_i(x_i^j, u_i, e_i^j) & i=0,\ldots N-1 \ ; \ j=1,\ldots r \\ x_o \quad \text{donné} \end{cases}$$

Remarque 3

Dans certain problème, on veut réaliser physiquement le Feedback. La loi de Feedback nous est alors donnée

$$(6) \qquad u_i^j = g_i(x_i^j, a_i)$$

Le problème consiste à déterminer les paramètres a_i optimaux.

En remplaçant (6) dans (1) on obtient le problème :

$$(7) \qquad x_{i+1}^j = f_i(x_i^j, g_i(x_i^j, a_i), e_i^j) \quad i=0,\ldots N-1 \ ; \ j=1,\ldots r$$

$$(8) \qquad p_i(x_i^j, g_i(x_i^j, a_i)) \leqslant 0 \quad i=0,\ldots N-1 \ ; \ j=1,\ldots r$$

$$(9) \qquad q_i(g_i(x_i^j, a_i)) \leqslant 0 \quad i=0,\ldots N-1 \ ; \ j=1,\ldots r$$

$$(10) \qquad \min_a \frac{1}{r} \sum_{j=1}^{r} \left(\sum_{i=0}^{N-1} L_i(x_i^j, g_i(x_i^j, a_i)) + M(x_N^j) \right)$$

où $a = (a_o, a_1, \ldots a_{N-1})$

Nous avons ainsi obtenu un problème open loop.

§ 2 - ALGORITHMES DE RESOLUTION DES PROBLEMES P_r, O_r, R_r.

Une fois les réalisations du bruit obtenues, nous sommes donc ramenés à un problème de contrôle déterministe avec contraintes sur l'état.

Par des méthodes de pénalisation ou de dualisation nous obtiendrons à un problème sans contraintes sur l'état.

Pour résoudre les problèmes ainsi obtenus nous exposerons les algorithmes de gradient et d'UZAWA. Ces méthodes nécessitant l'obtention du gradient, nous le donnerons explicitement pour nos trois problèmes.

1. Pénalisation extérieure pour une étude dans le cas convexe LIONS-TREMOLIERES-GLOWINSKI (6) . Soit le problème

$$(11) \qquad \begin{cases} \text{Inf } G(u) \\ u \in K \subset \mathbb{R}^m \end{cases}$$

faisons les hypothèses :

(i) - G est une fonction continue sur $\mathbb{R}^m$,

(ii) - $K = \{u \in \mathbb{R}^m \; h_p(u) \leq 0 \quad 1 \leq p \leq q \quad h_p : \mathbb{R}^m \to \mathbb{R} \text{ continue}\}$

supposons $K \neq \emptyset$

(iii) - ($G(v) \to +\infty$ lorsque $\|v\| \to +\infty \qquad v \in \mathbb{R}^m$) ou K borné.

Le problème pénalisé s'écrira

$$\inf_{u \in \mathbb{R}^m} \quad E(u,\varepsilon,\alpha) = G(u) + \frac{1}{\varepsilon} \sum_{p=1}^{q} \left(h_p^+(u)\right)^\alpha \qquad \alpha > 0$$

où $\varepsilon > 0$

$h_p^+(u) = \sup\,(h_p(u), 0)$

Remarque 1

Soit $K_M = \{v : G(v) \leq M\}$

K_M est compact en effet :

- K_M est fermé, car G continue
- K_M est borné grâce à l'hypothèse (iii).

Soit $V = \{v : G(v) = \text{Inf } G(u) \quad u \in K\}$

V n'est pas vide car d'après (iii)

$$\exists M_o : K_{M_o} \cap K \neq \emptyset$$

Or K est fermé, K_{M_o} est compact donc $K \cap K_{M_o}$ est compact. Comme G est continue on a le résultat.

D'autre part

$$E(u,\varepsilon,\alpha) \begin{cases} = G(u) & \text{si } u \in K \\ \geqslant G(u) & \text{si } u \notin K \end{cases}$$

possède les propriétés (i) (iii) donc

$$V(\varepsilon,\alpha) = \{ v : E(v,\varepsilon,\alpha) = \inf_u E(u\varepsilon,\alpha) \} \neq \emptyset$$

on a alors en posant $\varepsilon = \frac{1}{n}$

Théorème 1

$$\overline{\lim \sup \; V(\tfrac{1}{n},\alpha)} \subset V$$

Démonstration

Soit $v_n \in V(\frac{1}{n},\alpha)$ et $v \in V$ on a :

$$G(v_n) \leqslant E(v_n,\tfrac{1}{n},\alpha) \leqslant \inf_{u \in K} E(u,\tfrac{1}{n},\alpha) = \inf_{u \in K} G(u) = G(v)$$

v_n appartient donc à un borné de $\mathbb{R}^m$, on peut donc extraire une sous suite $v_{n'}$ de v_n convergente.

$$v_{n'} \longrightarrow w \in \overline{\lim \sup \; V(\tfrac{1}{n},\alpha)}$$

on a

$$G(w) \leqslant G(v)$$

donc

$$n' \left(h_p^+(v_{n'})\right)^\alpha \leqslant G(v) - G(v_{n'})$$

d'où

$$h_p^+(w) = 0 \quad \text{donc} \quad w \in K \quad \text{donc} \quad w \in V$$

Remarque 2

Supposons h_p et G dérivables alors pour $\alpha > 1$ E est dérivable par rapport à u.

Théorème 2

Si G est strictement convexe et h_p convexes $p=1,\ldots q$ $V(\frac{1}{n},\alpha)$ (resp V) contiennent un élément unique v_n (resp v) et on a :

$$\lim_{n\to\infty} v_n = v$$

Démonstration LIONS-TREMOLIERES-GLOWINSKI (6)

Application aux problèmes P_r, O_r, R_r .

Nous pénaliserons les contraintes faisant intervenir l'état du système :

- Les contraintes (2) et (4) pour le problème P_r .
- La contrainte (2) pour les problèmes R_r et O_r.

2. Méthodes duales

Démonstration dans

LIONS-TREMOLIERES-GLOWINSKI (6) ou BENSOUSSAN-LIONS-TEMAM (7)

2.1. Dualisation des contraintes sur l'état.

Soit le problème :

$$(12)\qquad \begin{cases} \min\ G(u) \\ u \in C \subset \mathbb{R}^m \\ h(u) \leqslant 0 \end{cases} \qquad h : \mathbb{R}^m \to \mathbb{R}^q$$

- $\cdot$ désignera le produit scalaire dans $\mathbb{R}^q$

Introduisons le Lagrangien

$$(13)\qquad L(u,\lambda) = G(u) + \lambda.h(u)$$

définition 1

Le couple (u^o,λ^o) $u^o \in C$ $\lambda^o \geqslant 0$ sera appelé point - selle s'il vérifie

$$L(u^o,\lambda) \leqslant L(u^o,\lambda^o) \leqslant L(u,\lambda^o) \qquad \forall u \in C,\ \forall \lambda \geqslant 0$$

Théorème 1

Si (u^o,λ^o) est un point-selle alors u^o est un minimum du problème(12).

Théorème 2

(Existence d'un point de selle).

Si

(i) C convexe fermé

(ii) G, h_p $p=1,\ldots q$ convexe sur C, G s.c.i.

(iii) $\exists u \in C$ $h_p(u) < 0$ $p=1,\ldots q$ (hypothèse de qualification)*

alors il existe un point - selle si et seulement si (12) admet une solution.

Le problème (12) se ramène donc à la recherche d'un point - selle du Lagrangien

Application

Nous dualiserons les contraintes sur l'état (2) des problèmes R_r et O_r .

2.2. Décomposition

Soit le problème d'optimisation global

(*)on peut remplacer (iii) par : $\exists$ une sphère de centre O et de rayon a dans $\mathbb{R}^q$: $\forall \psi$ $|\psi| = a$ l'équation $h(u) = \psi$ possède une solution dans C

$$(14)\qquad \begin{cases} \underset{v^1,\ldots,v^r}{\text{Min}} \quad \sum_{j=1}^{r} G^j(v^j) \\ v^j \in C^j \subset \mathbb{R}^m \\ \sum_{j=1}^{r} B^j(v^j) = 0 \quad \text{où} \quad B^j : \mathbb{R}^m \to \mathbb{R}^n \end{cases}$$

On définit les problèmes d'optimisation locaux

$$(15)\qquad \begin{cases} \underset{v^j}{\text{Min}} \quad G^j(v^j) + q.B^j(v^j) \\ v^j \in C^j \end{cases}$$

Définition 2

On dira que les problèmes locaux sont coordonnables s'il existe $p \in \mathbb{R}^n$ tel que u solution du problème global (14) soit solution des problèmes locaux pour $q=p$.

Théorème 3

Une CNS pour que les problèmes locaux soient coordonnables est que la fonctionnelle

$$(16)\qquad \sum_{j=1}^{r} G^j(v^j) + q.\left(\sum_{j=1}^{r} B^j(v^j)\right)$$

possède un point-selle sur $q \in \mathbb{R}^n$; $v^j \in C^j$, $j=1,\ldots r$

Théorème 4

Supposons B^j affines $j=1,\ldots r$, G convexe, s.c.i., inf. compacte alors, (16) a un point-selle.

Application

Les problèmes R_r (resp O_r) peuvent être considérés comme r problèmes d'optimisation couplés par la contrainte linéaire (4') (resp(4")).

Nous utiliserons cette méthode pour résoudre R_r .

2.3. Algorithme de gradient (pour une étude systématique gradient, gradient projeté, gradient conjugué LIONS-TREMOLIERES-GLOWINSKI (6))

Soit le problème :

$$(17) \qquad \begin{cases} \text{Inf } G(v) \\ v \in C \subset \mathbb{R}^m \end{cases}$$

C convexe fermé,

l'algorithme du gradient avec projection sera défini par la récurrence :

$$u^o \text{ donné}$$

$$u^{n+1} = P_C \left(u^n - \rho^n \nabla G(u^n) \right)$$

où

- P_C désigne la projection sur C
- ∇G désigne le gradient de G
- $0 < \rho_o < \rho^n < \rho_1$ ρ_o, ρ_1 pouvant être calculé explicitement (6).

Remarque 1

-A chaque itération il faut faire une projection sur C. On aura donc intérêt à ce que C soit de forme "simple" par exemple

$$C = \{ u : \alpha_i \leq u_i \leq \beta_i \qquad i=1,\ldots m \}$$

-Dans les problèmes P_r, R_r, O_r les contraintes sur l'état (2) sont difficiles à manier. C'est la raison pour laquelle nous les avons dualisées ou pénalisées.

Théorème

Si

(i) G strictement convexe de classe C_1

(ii) $\| v \| \to \infty \Rightarrow \quad G(v) \to + \infty$

(iii) C convexe fermé.

il existe ρ_o et ρ_1 tel que $0 < \rho_o < \rho^n < \rho_1$ entraîne $u^n \to u$ solution de (17)

Démonstration

G de classe C_1 entraîne G lipschitzien sur tout compact.

La démonstration de LIONS-TREMOLIERES-GLOWINSKI (6) s'adapte immédiatement.

Remarque 2

- On pourra prendre ρ^n optimum réalisant

$$\min_{\rho} \quad G\left(P_C(u^n - \rho \nabla G(u^n))\right)$$

2.4. Algorithme d'UZAWA

(Recherche d'un point de selle)

Reprenons les notations du 2

Faisons les hypothèses.

(i) G strictement convexe de classe C_1.

(ii) $G(v) \to +\infty$ lorsque $\|v\| \to +\infty$

(iii) $h_p(v)$ $p=1,\ldots q$ convexes

(iv) $\|h(v) - h(v')\| \leqslant C \|v-v'\|$ $v,v' \in \mathbb{R}^m$

(v) le lagrangien $G(v) + \lambda\, h(v)$ possède un point selle.

Alors on définit l'algorithme par la récurrence

(I) λ_o donné

(II) $\lambda^n \to v_{\lambda^n}$ réalisant $\min_v \left(G(v) + \lambda . h(v)\right)$

(III) $\lambda^{n+1} = P_\Lambda \left(\lambda^n + \rho^n h(v_{\lambda^n}) \right)$

où

P_Λ désigne la projection sur $\Lambda = \{ \lambda_p \geqslant 0 \quad p=1,\ldots q \}$ $\rho_o < \rho^n < \rho_1$ et

ρ_o et ρ_1 sont deux constantes fixes voir (7)

Remarque 2

L'étape (II) sera réalisée par un algorithme de minimisation classique par exemple par l'algorithme de gradient.

Théorème

Sous les hypothèses (i).....(vi) on a :

$v_{\lambda^n} \longrightarrow u$ solution du problème initial (12)

$G(v_{\lambda^n}) \longrightarrow G(u)$ coût minimum de (12)

λ^n appartient à un borné ; Toute suite extraite converge vers λ, et (u,λ) est point — selle du lagrangien.

démonstration : voir BENSOUSSAN-LIONS-TEMAM (7)

Remarque 3

L'algorithme d'UZAWA est l'algorithme de gradient appliqué à la fonction

$$H(\lambda) = \min_{v \in C} \left(G(v) + \lambda . h(v) \right)$$

En effet supposons H différentiable

$$H(\lambda) = G(v_\lambda) + \lambda . h(v_\lambda)$$

$$H'(\lambda) = G'_v . v'_\lambda + h_p(v_\lambda) + \lambda . h'_v . v'_\lambda$$

Or v_λ vérifie

$$G'(v_\lambda) + \lambda . h'_v = 0$$

donc $H'_\lambda = h_p(v_\lambda)$

2.5. Calcul des gradients des problèmes P_r, R_r, O_r .

Nous donnerons dans ce paragraphe les formules définissant les gradients des problèmes P_r, R_r, O_r (les contraintes sur l'état seront pénalisées).

Pour l'étape de minimisation de l'algorithme d'UZAWA on a besoin du gradient des problèmes P_r, R_r, O_r sans contraintes sur l'état (Il suffira d'annuler les termes de pénalisation dans les gradients précédents).

2.5.1. Calcul du gradient pour un problème de contrôle à temps discret

Soit le problème de contrôle :

$$(18)\qquad \begin{cases} x_{i+1} = f_i(x_i,u_i) \quad i=0,\dots N-1 \ ; \quad \text{où} \quad f_i : \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}^n \\ \\ x_o \quad \text{donné} \end{cases}$$

$$(19)\qquad \text{Min} \quad G(u) = \sum_{i=0}^{N-1} L_i(x_i,u_i) + M(x_N).$$

Faisons les hypothèses :

L_i, M, f_i $i=0,\dots N-1$ continument dérivables. Si δ_x est l'accroissement de x dû à δu de (18) on obtient

$$(20)\qquad \begin{cases} \delta x_{i+1} = f'_{i_{x_i}} . \delta x_i + f'_{i_{u_i}} . \delta u_i \\ \\ \delta x_o = 0 \end{cases}$$

d'autre part on a de (19)

$$(21)\qquad G_u . \delta u = \sum_{i=0}^{N-1} (L'_{i_{x_i}} . \delta x_i + L'_{i_{u_i}} . \delta u_i) + M'_{x_N} . \delta x_N$$

Définissons alors le système dual par :

$$(22)\qquad \begin{cases} \lambda_i = \lambda_{i+1} \cdot f'_{i_{x_i}} - L'_{i_{x_i}} \\ \\ \lambda_N = - M'_{x_N} \end{cases}$$

$\lambda_{i+1} . (20) - (22) . \delta_{x_i}$ donne

$$(23)\qquad \sum_{i=0}^{N-1} L'_{i_{x_i}} \cdot \delta x_i + M'_{x_N} \cdot \delta x_N = - \sum_{i=0}^{N-1} \lambda_{i+1} \cdot f'_{i_{u_i}} \cdot \delta u_i$$

(23) et (21) donne

$$G_u \cdot \delta u = \sum_{i=0}^{N-1} (L'_{i_{u_i}} - \lambda_{i+1} f'_{i_{u_i}}) \cdot \delta u_i$$

d'où le gradient

$$(L'_{i_{u_i}} - \lambda_{i+1} \cdot f'_{i_{u_i}}) \quad i=0,\ldots N-1)$$

2.5.2. Application au problème P_r pénalisé

posons

$$\mathcal{L}(\varepsilon,i,j) = L_i(x_i^j,u_i^j) + \frac{1}{\varepsilon} p_i^+(x_i^j,u_i^j)^\alpha$$

$$\mathcal{J}(\varepsilon,i,j,j') = \frac{1}{\varepsilon} \left((|u_i^j - u_i^{j'}| - K|x_i^j - x_i^{j'}|)^+\right)^\alpha$$

le problème P_r pénalisé s'écrit alors

$$(24)\qquad \begin{cases} x_{i+1}^j = f_i(x_i^j,u_i^j,e_i^j) & i=0,\ldots N-1 \ ; \ j=1,\ldots r \\ x_o \quad \text{donné} \end{cases}$$

$$(25)\qquad q_i(u_i^j) \leqslant 0$$

$$\min_u G(u) = \sum_{j=1}^{r} \left\{ \sum_{i=0}^{N-1} \left(\mathcal{L}(\varepsilon;i,j) + \sum_{j \neq j'} \mathcal{J}(\varepsilon,i,j,j') \right) + M(x_N^j) \right\}$$

Remarque 1

Nous avons multiplié le critère par r afin de simplifier les écritures. Nous ferons de même pour les problèmes R_r et O_r.

Le système dual :

$$(26)\qquad \begin{cases} \lambda_i^j = \lambda_{i+1}^j \cdot f_{i\,x_i^j} - \mathscr{L}'_{j\,x_i^j}(\varepsilon,i,j) - \sum_{j'\neq j} \mathscr{P}'_{j\,x_i^j}(\varepsilon,i,j,j') & i=0,\ldots,N-1 \\ & j=1,\ldots,r \\ \lambda_N^j = -M'_{x_N^j} \end{cases}$$

Les composantes du gradient correspondantes à u_i^j sont alors

$$(27)\qquad \mathscr{L}'_{j\,u_i^j}(\varepsilon,i,j) + \sum_{j\neq j'} \mathscr{P}'_{j\,u_i^j}(\varepsilon,i,j,j') - \lambda_{i+1}^j . f'_{i\,u_i^j}$$

Remarque 2

Les formules (26)(27) explicitent les couplages existant entre les diverses réalisations. Ce sont les termes en $\mathscr{P}$.

2.5.3. Application au problème R_r pénalisé

Le problème R_r s'écrit

$$(28)\qquad \begin{cases} \left.\begin{cases} x_1^j = f_o(x_o,u_o,e_o^j) \\ x_{i+1}^j = f_i(x_i^j,u_i^j,e_i^j) \qquad i=1,\ldots N-1 \end{cases}\right\} j=1,\ldots r \\ x_o \text{ donné} \end{cases}$$

$$(29)\qquad q_i(u_i^j) \leqslant 0 \qquad i=0,\ldots N-1 \;;\; j=1,\ldots r$$

$$(30)\qquad \min_u G(u) = \sum_{j=1}^{r} \{ \sum_{i=0}^{N-1} \mathscr{L}(\varepsilon,i,j) + M(x_N^j) \}$$

Le système dual

$$(31)\qquad \begin{cases} \lambda_i^j = \lambda_{i+1}^j \cdot f'_{i\,x_i^j} - \mathcal{L}'_{x_i^j}(\varepsilon,i,j) \\ \lambda_N^j = -M'_{x_N^j} \end{cases}$$

Notons

$$(32)\qquad \ell_u(i,j) = \mathcal{L}'_{u_i^j}(\varepsilon,i,j) - \lambda_{i+1}^j \cdot f'_{i\,u_i^j}$$

Le gradient vaut alors

$$(33)\qquad \begin{cases} i\neq 0 \quad g_{u_i^j} = \ell_u(i,j) \\ i=0 \quad g_{u_o} = \sum_{j=1}^{r} \ell_u(i,j) \end{cases}$$

Remarque 3

$\ell_u(i,j)$ est le gradient du problème de minimisation associé à la seule réalisation j

$$(34)\qquad \begin{cases} x_i^j = f(x_i^j, u_i^j, e_i^j) \quad i=0,\ldots N-1 \qquad j \text{ fixé} \\ x_o \quad \text{donné} \end{cases}$$

$$(35)\qquad q_i(u_i^j) \leq 0 \quad i=0,\ldots N-1 \qquad j \text{ fixé}$$

$$(36)\qquad \min_{(u_i^j,\, i=0,\ldots N-1)} g^j(u) = \sum_{i=0}^{N-1} \mathcal{L}(\varepsilon,i,j) + M(x_N^j)$$

On peut donc découpler le problème d'optimisation (28), (29), (30) en r problèmes (34), (35), (36). L'algorithme de gradient peut alors se décomposer de la façon suivante :

(I) Calculer $\ell_u(i,j)$ gradient du problème (34), (35), (36) pour $j=1,\ldots,r$

(II) Le gradient du problème R_r est alors donné par (33).

Cet algorithme de décomposition sera qualifié de naturel par opposition à l'algorithme d'UZAWA associé au lagrangien (16) de 2.2.

2.5.4. Application au problème O_r pénalisé

Soit le problème O_r

$$(37)\qquad \begin{cases} x_i^j = f_i(x_i^j, u_i, e_i^j) & i=0,\ldots N-1 \ ; \ j=1,\ldots r \\ x_o \ \text{donné} \end{cases}$$

$$(38)\qquad q_i(u_i) \leq 0$$

$$(39)\qquad \min_u G(u) = \sum_{j=1}^{r} \{ \sum_{i=0}^{N-1} \mathcal{L}(\varepsilon,i,j) + M(x_N^j) \}$$

Le système dual

$$(40)\qquad \begin{cases} \lambda_i^j = \lambda_{i+1}^j . f'_{i\,x_i^j} - \mathcal{L}'_{x_i^j}(\varepsilon,i,j) \\ \lambda_N^j = - M'_{x_N^j} \end{cases}$$

Soit

$$(41)\qquad \ell_u(i,j) = \mathcal{L}'_{u_i^j}(\varepsilon,i,j) - \lambda_{i+1}^j . f'_{u_i^j}$$

Remarque 4

Les systèmes duaux des problèmes O_r et R_r ont la même forme. On peut faire une remarque analogue à la remarque 3. On peut décomposer l'algorithme de gradient.

(I) Calculer $\ell_u(i,j)$ gradient de (34), (35), (36)

(II) le gradient de O_r est alors donné par (42).

§ 3 - LES METHODES DE REFERENCE.

Nous exposerons brièvement ici les méthodes de la programmation dynamique et de KALMAN. Elles nous permettront en effet d'apprécier la qualité des résultats obtenus en résolvant les problèmes P_r, R_r et O_r.

3.1. Programmation dynamique classique.

Soit à résoudre :

$$\text{(43)} \qquad \begin{cases} x_{i+1} = f_i(x_i, u_i, e_i) & i=0,\dots N-1 \\ x_o \text{ donné} \end{cases}$$

$$\text{(44)} \qquad x_i \in C_i \subset \mathbb{R}^m \qquad i=1,\dots N$$

$$\text{(45)} \qquad u_i \in D_i \qquad i=0,\dots N-1$$

$$\text{(46)} \qquad \min_u \; E\Big\{ \sum_{i=0}^{N-1} L_i(x_i,u_i) + M(x_N) \Big\}$$

Appelons

$$\text{(47)} \qquad V_j(x_j) = \min_{u_j,\dots,u_{N-1}} E\Big(\sum_{i=j}^{N-1} L_i(x_i,u_i) + M(x_N) \Big)$$

x_j donne

(43)

(44)

(45)

Le principe de BELLMAN donne alors la récurrence définissant $V=(V_o,\dots V_N)$

$$\text{(48)} \qquad \begin{cases} V_i(x_i) = \min_{u_i \in D_i} E\{L_i(x_i,u_i) + V_{i+1}(f_i(x_i,u_i,e_i))\} \; ; \; x_i \in L_i \\ V_N(x_N) = M(x_N) \quad ; \quad x_N \in C_N \end{cases}$$

A) Supposons e_i, u_i $i=0,\ldots,N-1$ discret alors l'algorithme est immédiat.

On calcule par récurrence les $V_i(x_i)$ en calculant pour chaque valeur de u_i :

$E\{ L_i(x_i,u_i) + V_{i+1}(x_{i+1}) \}$.

B) Supposons e_i, u_i continus

- (I) on discrétise la loi de Probabilité des e_i en tirant des réalisations de e_i^j par exemple.
- (II) On calcule au point x_i $V_i(x_i)$ en minimisant $\mathcal{V}_i^r(x_i,u_i)$ par rapport à u_i où :

$$\mathcal{V}_i^r(x_i,u_i) = \frac{1}{r} \sum_{j=1}^{r} \left(L_i(x_i,u_i) + V_{i+1}(x_{i+1}^j)\right)$$

Remarque 1

Nous ne pourrons calculer $V_i(x_i)$ que pour un ensemble discret de x_i. $V_i(x)$ sera alors déterminé en tout point par interpolation linéaire.

Remarque 2

Dans l'étape (II) :

a) si $\mathcal{V}_i^r(x_i,u_i)$ n'est pas convexe par rapport à u_i il faudra discrétiser l'ensemble D_i et calculer $\mathcal{V}_i^r$ pour toutes les valeurs u_i .

b) Si $\mathcal{V}_i^r(x_i,u_i)$ est convexe on pourra utiliser une méthode de gradient, ou faire une "dichotomie" utilisant les nombres de Fibonacci (suivant la valeur de m).

L'algorithme (B) est donné dans JACOBSON et MAYNE [11]

Théorème 1

Si f_i linéaire, L_i convexe globalement par rapport à x et u $\forall i=0,\ldots,N-1$ et si M convexe alors la fonction de Bellman est convexe par rapport à x .

démonstration immédiate par récurrence.

Sous les hypothèses du Théorème 1 on est dans le cas de la remarque 2b.

Remarque 3

Dés que la dimension de l'état est supérieure à 3 cette méthode devient inutilisable.

Remarque 4

Utilisant le chapitre I, sous certaines hypothèses de continuité, on peut montrer la convergence de l'algorithme (B).

3.2. La Méthode de KALMAN.

Dans ce paragraphe nous ferons les hypothèses suivantes

(i) Equation d'évolution du système linéaire,

(ii) Coût quadratique,

(iii) Bruits gaussiens indépendants,

(iiii) Observation bruitée.

Ce problème sera appelé Linéaire Quadratique Gaussien (LQG)

L'équation d'évolution

$$(49) \qquad \begin{cases} x_{i+1} = \phi_i x_i + \psi_i u_i + \Gamma_i e_i & i=0,\dots N-1 \\ x_o \text{ gaussienne centrée} \end{cases}$$

L'observation

$$(50) \qquad z_{i+1} = H_{i+1} x_{i+1} + f_{i+1} \qquad i=0,\dots N-1$$

e_i, f_{i+1} $\quad i=0,\dots N-1$ indépendantes gaussiennes centrées

$$(51) \qquad \min_u \; E\Big\{ \sum_{i=0}^{N-1} \left({}^t x_i A_i x_i + {}^t u_i B_i u_i \right) + {}^t x_N A_N x_N \Big\}$$

Nous cherchons u parmi les stratégies feedback fonction uniquement des observations passées

$$u_i = h_i(z_1, \dots z_i)$$

Notons $\hat{x}(k|i) = E(x_k|z_1,\ldots,z_i)$

On a

Théorème 2

(de séparation)

La stratégie optimale du problème (49) (50) (51) est $u_k = S_k.\hat{x}(k|k)$ où S_k est la matrice donnant le Feedback optimal $S_k.x_k$ du problème déterministe (49), (51) où $e_i = 0 \quad i=0,\ldots,N-1$

Démonstration voir par exemple MEDITCH (8) .

Remarque 5

Sous des hypothèses de convexité WONHAM (9) donne des résultats analogues pour les problèmes de contrôle en temps continu. BENSOUSSAN (10) donne le résultat, lorsque le système est régi par des équations aux dérivées partielles.

Remarque 6

Le théorème de séparation est très important car il montre que, sous certaines hypothèses, on peut décomposer la recherche du Feedback optimal en

- Un problème de filtrage,
- Un problème de contrôle déterministe.

Remarque 7

Nous aurons à utiliser ce théorème dans les cas où $f_{i+1} = 0$, $H_{i+1} = I$ matrice unité (observatior complète).

La résolution du problème P_1 pour la réalisation $e_i = 0$ donnera la commande initiale exacte. Plus généralement il suffira sous les hypothèses (i) (ii) (iii) (iiii) de générer des réalisations du bruit symétrique pour obtenir exactement la commande initiale.

Remarque 8

Soit $\mathcal{U}_i$ l'ensemble de stratégies suivant :

$$\mathcal{U}_i = \left\{ (a_o,\ldots,a_N) \quad \begin{array}{ll} j \leq i & a_j \text{ est fonction de } z_1,\ldots,z_j \\ j > i & a_j \text{ est fonction de } z_1,\ldots,z_i \end{array} \right\}$$

cherchons la stratégie optimale du problème LQG dans la classe $\mathcal{U}_i$ on obtient le :

Théorème 3

La stratégie optimale du problème LQG dans $\mathcal{U}_i$ est donnée par

$$u_k = S_k \cdot \hat{x}(k|i)$$

où S_k définit la stratégie feedback $S_k.x_k$ du problème déterministe (49), (51), $e_j = 0$, $j=0,\ldots,N-1$

démonstration : Analogue au théorème de séparation.

Corollaire 1

Les stratégies Feedback et Open loop coïncident à l'instant 0 dans le cas LQG

Définition

Stratégie OLFO (Open loop Feedback optimal) TSE (13)

Elle sera définie par la récurrence suivante. Supposons la stratégie connue jusqu'à l'instant $i-1$ $(\tilde{a}_o,\ldots,\tilde{a}_{i-1})$ a_i sera obtenue en minimisant le problème dans la classe $\mathcal{U}_i\{ a_o=\tilde{a}_o,\ldots,a_{i-1}=\tilde{a}_{i-1} \}$.

Corollaire 2

Pour les problèmes LQG la stratégie OLFO coïncide avec la stratégie Feedback optimale.

§ 4 - LE PROBLEME $P(\mu_r,\alpha_r)$

Dans ce paragraphe nous ferons quelques remarques sur les problèmes où apparaissent des contraintes en probabilité.

Nous appellerons P_r^α le problème $P(\mu_r(\omega),\alpha_r)$ associé à la classe de stratégies $\mathcal{Y}_k$ définie au §1.1

$$(52)\qquad \begin{cases} x_{i+1}^j = f_i(x_i^j,u_i^j,e_i^j) & i=0,..,N-1 \ ; \ j=1,...,r \\ x_o \text{ donné} \end{cases}$$

$$(53)\qquad \frac{1}{r}\,\text{card}\{\, j : p_i(x^j) \leqslant 0\} \geqslant \alpha - \varepsilon_r \quad i=1,...,N \ ; \ j=1,...,r$$

$$(54)\qquad q_i(u_i^j) \leqslant 0 \qquad i=0,...,N-1 \ ; \ j=1,...,r$$

$$(55)\qquad |u_i^j - u_i^{j'}| \leqslant k|x_i^j - x_i^{j'}| \qquad i=0,...,N-1 \ ; \ j\neq j' \quad j=1,...,r \quad j'=1,...,r$$

$$(56)\qquad \operatorname*{Min}_u \frac{1}{r} \sum_{j=1}^{r} \{ \sum_{i=0}^{N-1} L_i(x_i^j,u_i^j) + M(x_N^j) \}$$

où $\varepsilon_r = \dfrac{\lambda}{r^{\frac{1}{2}-\mu}}$ $\qquad \lambda > 0 \ , \quad \mu > 0$

On peut de même définir R_r^α et O_r^α en approximant (55) comme nous avons fait dans § 1.

La difficulté essentielle de ce type de problème est la contrainte (53).

Appelons J le plus petit entier $\geqslant (\alpha - \varepsilon_r)r$. On peut alors donner l'algorithme suivant :

(I) choisir $(j_1^i,...,j_J^i)$ J réalisations prises dans les r réalisations $(1,...,r)$ à l'instant i . Notons $j_k = (j_k^1,...,j^N)$

(II) Résoudre le problème du type P_r défini par (52) (54) (55) (56) et (57) avec

(57) $$p_i(x_i^{j_k^i}) \leqslant 0 \qquad \text{pour} \quad k=1,\ldots,J$$

Nous définissons ainsi les applications

- $(j_1,\ldots,j_J) \longrightarrow$ coût minimum du problème (52)(54)(55)(56)(57) $\mathcal{C}(j_1,..,j_J)$

- $(j_1,\ldots,j_J) \longrightarrow$ stratégie feedback optimale de (52)(54)(55)(56)(57) $S(j_1,\ldots,j_J)$.

La solution du problème P_r^α sera $S(j_1,\ldots,j_J)$ où $(j_1,\ldots,j_J)$ est la combinaison qui réalise le coût minimum $\mathcal{C}(j_1,..,j_J)$ parmi toutes les combinaisons possibles.

Remarque 1

Cet algorithme n'est utilisable que pour r faible, et J voisin de r ou de 1, car sinon le nombre de combinaison (C_r^{JN-1}) devient trop grand. Le coût de cet algorithme sera : $\sim (C_r^J)^{N-1}$ (le coût (temps de calcul) de P_r)

Remarque 2

Nous pourrons construire des algorithmes analogues pour les problèmes O_r^α et R_r^α .

Remarque 3

Pour certains problèmes nous n'avons pas besoin de la stratégie optimale, mais seulement d'une "bonne" stratégie admissible. On remplacera alors (53) par

(58) $$p_i'(x_i^j) \leqslant 0 \qquad \forall\ j = 1,\ldots,r \quad i=1\ \ldots,N$$

où p' est définie de telle façon que :

(59) $$p_i'(x_i) \leqslant 0 \ \text{ p.s. } \Rightarrow P\left(p_i(x_i) \leqslant 0\right) \geqslant \alpha$$

§ 5 - GENERATION DE REALISATION DE VARIABLES ALEATOIRES DE LOI DONNEE.

Nous avons supposé, jusqu'à présent, connaître les réalisations e_i^j du bruit e_i .

Nous donnons ici quelques méthodes pour les générer numériquement. Pour une étude systèmatique nous pourrons nous reporter à SHREIDER (12) .

5.1. Génération d'une variable aléatoire ξ uniformément répartie sur $(0,1)$.

Il existe de nombreuses méthodes nous donnons uniquement celles que nous avons utilisées.

ξ_n donné par (60) pourra être considéré comme un tirage de ξ .

(60) $$\xi_{n+1} = 7^{4k+1}\ \xi_n \quad \text{modulo } 10^{11}\ k \text{ fixé}$$

SHREIDER (12)

5.2. Génération d'une variable aléatoire gaussienne centrée réduite X .

Nous utiliserons le théorème central limite :

(61) $$X = \sqrt{2}\left(\sum_{j=1}^{6} \xi_j - 3\right)$$

où ξ_j sont des variables aléatoires uniformément réparties sur $(0,1)$ indépendantes entre elles.

5.3. Génération de la variable aléatoire Y de fonction de répartition F.

La variable aléatoire $Z = F(Y)$ est de loi uniforme sur $(0,1)$ en effet

$$z \in (0,1) \quad P(Z \leqslant z) = P\left(Y \leqslant F^{-1}(z)\right) = FF^{-1}(z) = z$$

$$z \leqslant 0 \quad (P\ Z \leqslant z) = 0$$

$$z \geqslant 1 \quad P(Z \leqslant z) = 1$$

Donc, soit ξ une v.a. uniformément répartie sur $(0,1)$; $Y=F^{-1}(\xi)$ sera une v.a. de fonction de répartition F .

CHAPITRE III

EXEMPLES NUMERIQUES

Nous donnons dans ce chapitre la solution de trois problèmes en résolvant les problèmes O_r , R_r et P_r associés.

Nous comparons les résultats obtenus aux solutions données par la programmation dynamique et par la méthode de KALMAN (lorsque c'est possible).

L'exemple 1 est purement formel. Les exemples deux et trois sont des Gestions de stock-production.

§-1 EXEMPLE I

SYSTEME LINEAIRE, COUT QUADRATIQUE, BRUIT GAUSSIEN

1. Le problème $P(\mu)$

$$(1) \qquad \begin{cases} x_{i+1} = x_i + 2u_i + \omega_i & i=0,1,2 \\ x_o \text{ donné} \end{cases}$$

où ω_i $i=0,1,2$ sont des lois gaussiennes centrées d'écart, type 10 ($\mathcal{N}(0,10)$), indépendantes.

Coût

$$(2) \qquad G(u) = E\{ \sum_{i=0}^{2} u_i^2 + x_3^2 \}$$

2. Solution Feedback théorique.

Le théorème de séparation (ch II§3.2) permet de calculer aisément la solution théorique. On obtient :

$$(3) \qquad \begin{cases} u_o = -0.154 \; x_o \\ u_1 = -0.222 \; x_1 \\ u_2 = -0.400 \; x_2 \end{cases}$$

Le coût optimal en fonction de x_o est donné par le tableau suivant

X_o	0	1	2	3	4	5	10
COUT	131.1	131.2	131.4	131.8	132.3	133.0	138.8

3. Solution Open loop théorique.

Utilisant le (Th3 chII§3.2) on obtient la solution open loop en fonction de la condition initiale

$$(4) \qquad \begin{cases} u_o = -0.154\, x_o \\ u_1 = -0.154\, x_o \\ u_2 = -0.154\, x_o \end{cases}$$

Le coût optimal open loop en fonction de x_o est donné par le tableau suivant

X_o	0	1	2	3	4	5	6
COUT	300.0	300.1	300.3	300.7	301.2	301.9	307.7

Donnons maintenant les résultats obtenus par les algorithmes de gradient appliqués aux problèmes O_r, R_r, P_r :

- La commande initiale optimale en fonction de l'état initial (pour 10, 20, 50 réalisations).
- Le coût optimal en fonction de l'état initial pour 10, 20, 50 réalisations
- L'influence de la classe de stratégie sur laquelle on minimise γ_k (chII §1.1) k=1, 10, 100.
- La qualité de la commande pour les instants postérieurs à l'instant initial.

4. Résolution de O_r (algorithme de gradient)

4.1. Temps de calcul en fonction du nombre de réalisations r sur CII 10 070

r	10	20	50
temps	$\sim$ 2s	$\sim$ 4s	$\sim$ 10s

4.2. Commande en fonction du nombre de réalisations r

nous obtenons $u_o = u_1 = u_2$

Le nombre de réalisations n'a pas d'influence. Nous obtenons exactement la solution théorique. Ceci s'explique :

- Utilisant la remarque (chI. §3.3) nous avons tiré les réalisations par couple symétrique.

- La commande optimale est linéaire en fonction de l'état initial (4)

x_o	0	1	2	3	4	5	10
$u_o=u_1=u_2$ r=10,20,50	0.00	-0.154	-0.308	-0.461	-0.615	-0.770	-1.54
$u_o=u_1=u_2$ théorique	0.00	-0.154	-0.308	-0.462	-0.615	-0.770	-1.54

4.3. Coût en fonction du nombre de réalisations r

x_o \ r	10	20	50	open loop théorique
0	179.5	237.1	270.1	300.0
1	179.6	237.2	270.2	300.1
2	179.8	237.4	270.4	300.3
3	180.2	237.8	270.8	300.7
4	180.7	238.3	271.3	301.2
5	181.4	339.0	272.0	301.9
10	187.2	244.8	277.8	307.7
Erreur relative	$\sim$ 0.4	$\sim$ 0.2	$\sim$ 0.1	

5. Résolution de R_r et de P_r .

Remarque 1

Les commandes initiales obtenues en résolvant les problèmes R_r et P_r sont identiques.

Les temps de calcul des problèmes R_r sont du même ordre que ceux de O_r.

5.1. Temps de calcul en fonction du nombre de réalisations r et de la classe $\mathcal{J}_k$ (ch II §1.1) CII 10 070

5.1.1. En fonction du nombre de réalisations.

r	10	20	50	
Temps	∿ 1mn	∿ 2mn	∿ 10mn	(P_r)

Remarque 2

On minimise sur $\mathcal{J}_1$

5.1.2. En fonction $\mathcal{J}_k$ pour 10 réalisations

k	1	10	100	
Temps	∿ 1mn	∿ 20s	∿ 15s	(P_r)

5.2. Commande initiale en fonction de :

- de l'état initial,
- du nombre de réalisations,
- de la classe de stratégie $\mathcal{S}_k$.

Le nombre de réalisations et la classe de stratégie n'ont pas d'influence sur la commande initiale u_o en fonction de x_o, obtenu par P_r

- pour r=10, 20, 50 sur $\mathcal{S}_1$
- pour r=10 sur $\mathcal{S}_1, \mathcal{S}_{10}, \mathcal{S}_{100}$.

X_o	0	1	2	3	4	5	10
u_o P_r	0.00	-0.154	-0.308	-0.461	-0.615	-0.769	-1.54
u_o théorique	0.00	-0.154	-0.308	-0.462	-0.615	-0.770	-1.54

5.3. Coût optimal en fonction de l'état initial.

5.3.1. En fonction du nombre de réalisations.

(On minimise dans $\mathcal{U}_1$)

x_o \ r	10	20	50	Feedback théorique
0	51	62.5	102.0	131.1
1	51.1	63.5		131.2
2	49.9	63.5		131.4
3	54.4	70.4		131.8
4	51.2	65.6		132.3
5	51.3	64.5		133.0
10	59.0	70.1	110.7	138.8
Erreur relative	$\sim$ 0,6	$\sim$ 0,5	$\sim$ 0,2	

5.3.2. En fonction de $\mathcal{J}_k$ pour 10 réalisations.

(Nous donnerons également le résultat de R_r pour 10 réalisations)

x_o \ $\mathcal{J}_k$	$\mathcal{J}_{100}$	$\mathcal{J}_{10}$	$\mathcal{J}_1$	$\mathcal{J}_{0.4}$	R_{10}	Feedback théorique
0	19.9	23.5	51.0	76.7	19.9	131.1
1	20.0	24.1	51.1	76.7	20.0	131.2
2	20.2	24.3	49.9	77.3	20.2	131.4
3	20.6	24.6	54.4	76.7	20.6	131.8
4	21.2	25.1	51.2	79.2	21.2	132.3
5	21.9	26.4	51.3	77.2	21.9	133.0
10	27.7	30.1	59.0	90.8	27.7	138.8

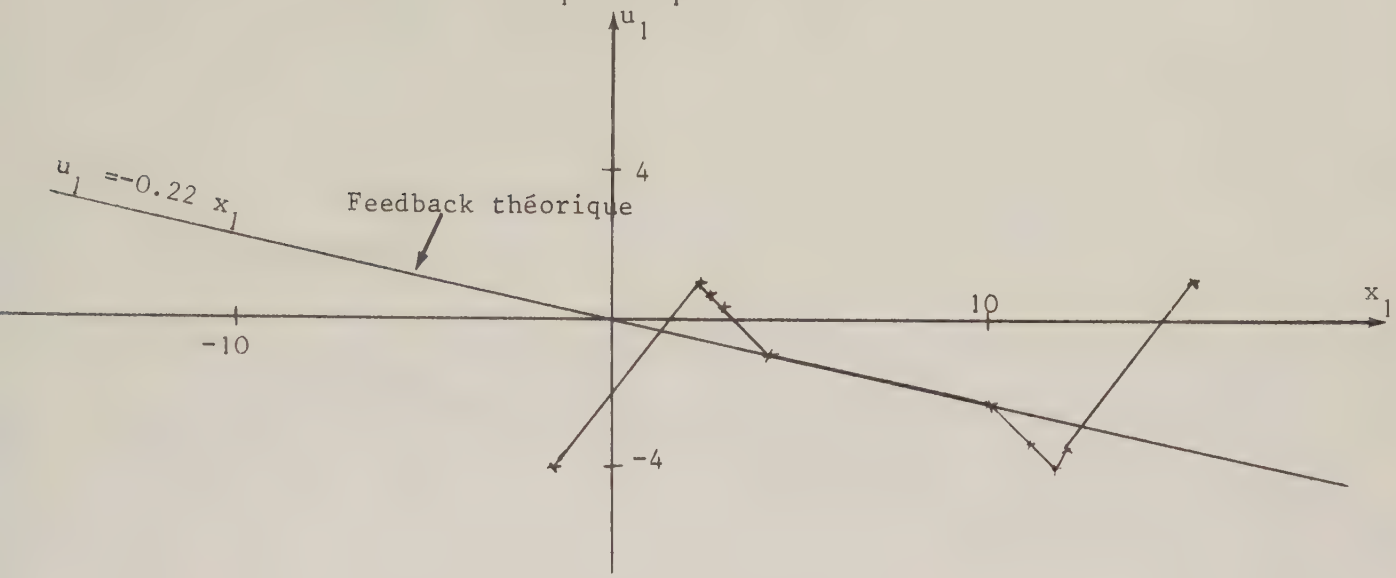
Remarque 3

Dans les tableaux 5.3.1. et 5.3.2. les perturbations des variations du coût en fonction de l'état initial proviennent de la non convexité des problèmes (ch II §1.2). Les algorithmes de gradient convergent vers des minimums locaux. Ces perturbations n'apparaissent pas pour les problèmes R_r et O_r car ils sont convexes.

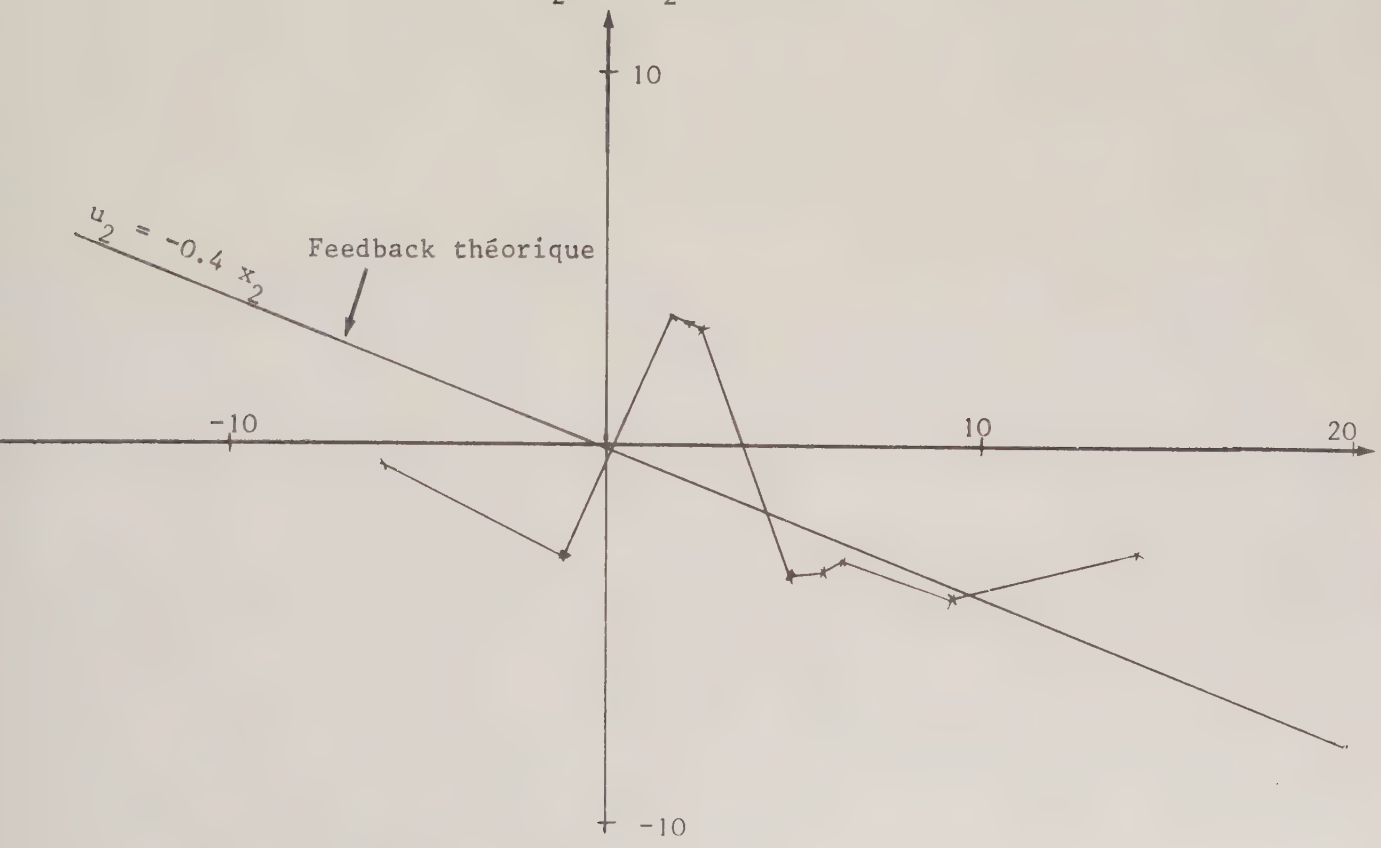
5.4. Commandes aux instants 2 et 3 en fonction de l'état (x_o =10)

5.4.1. P_r r=10 réalisations

5.4.1.1. $u_1 = f(x_1)$

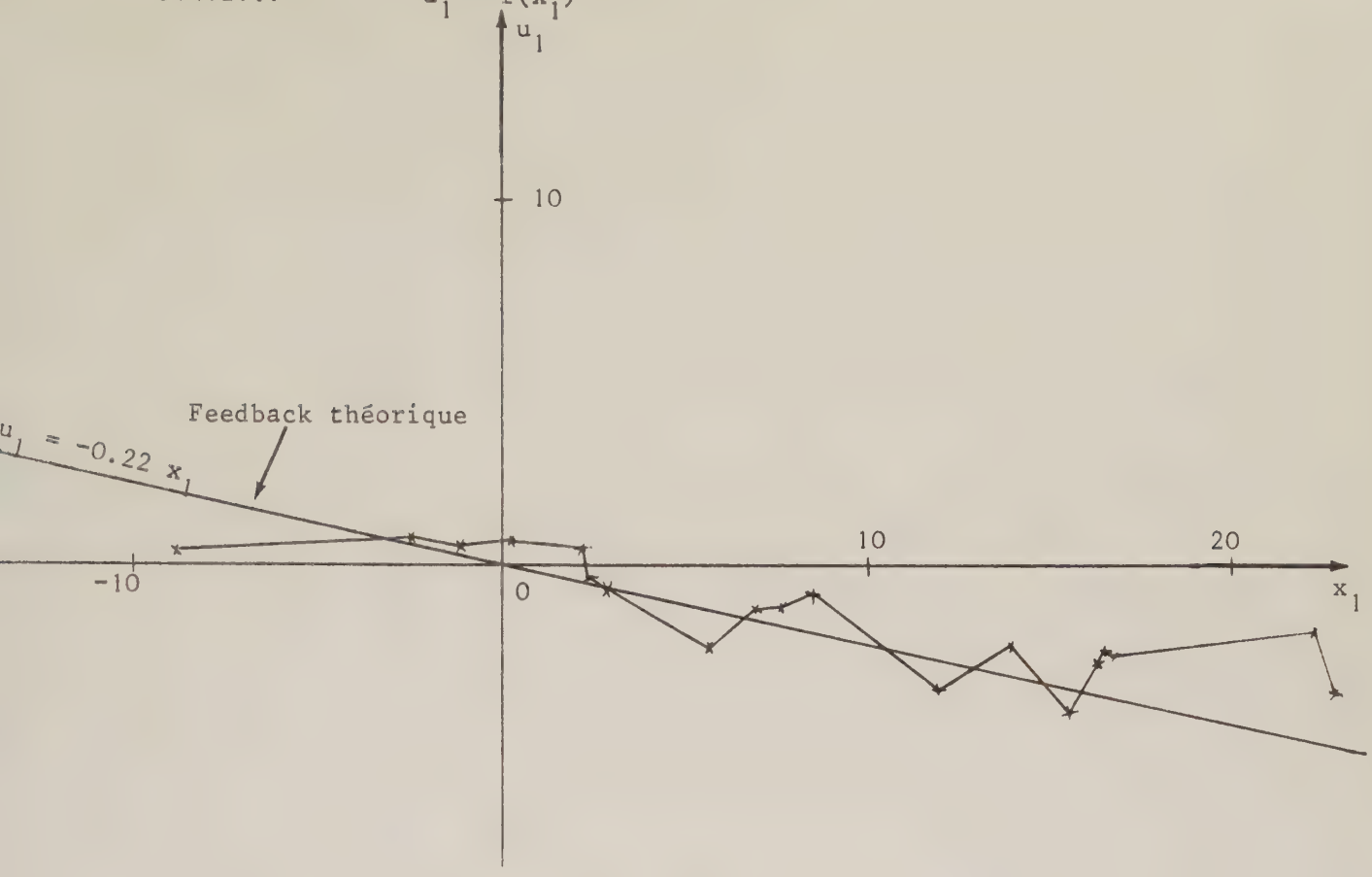


5.4.1.2. $u_2 = f(x_2)$

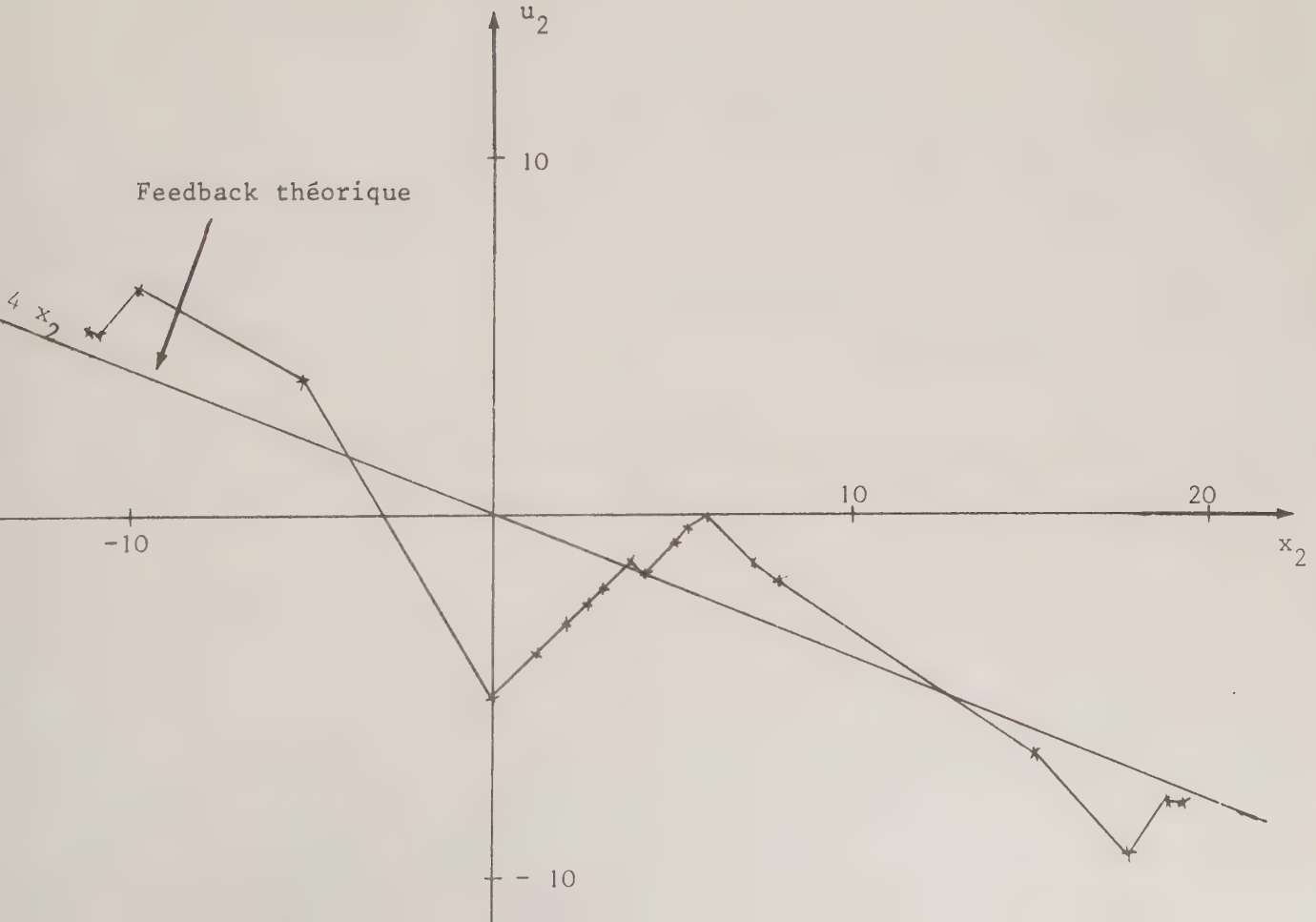


5.4.2.1. P_r 20 réalisations.

5.4.2.1. $u_1 = f(x_1)$

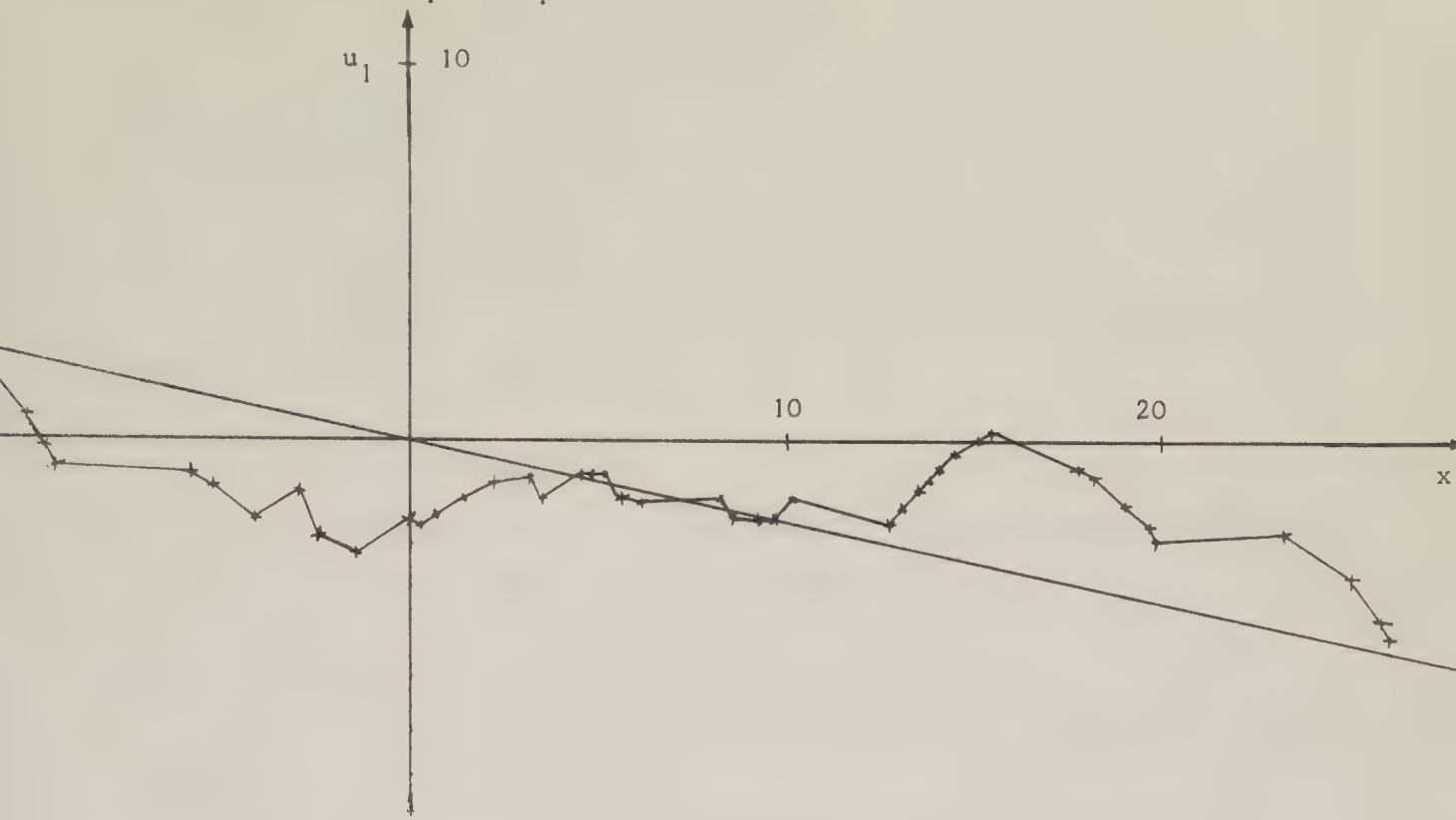


5.4.2.2. $u_2 = f(x_2)$

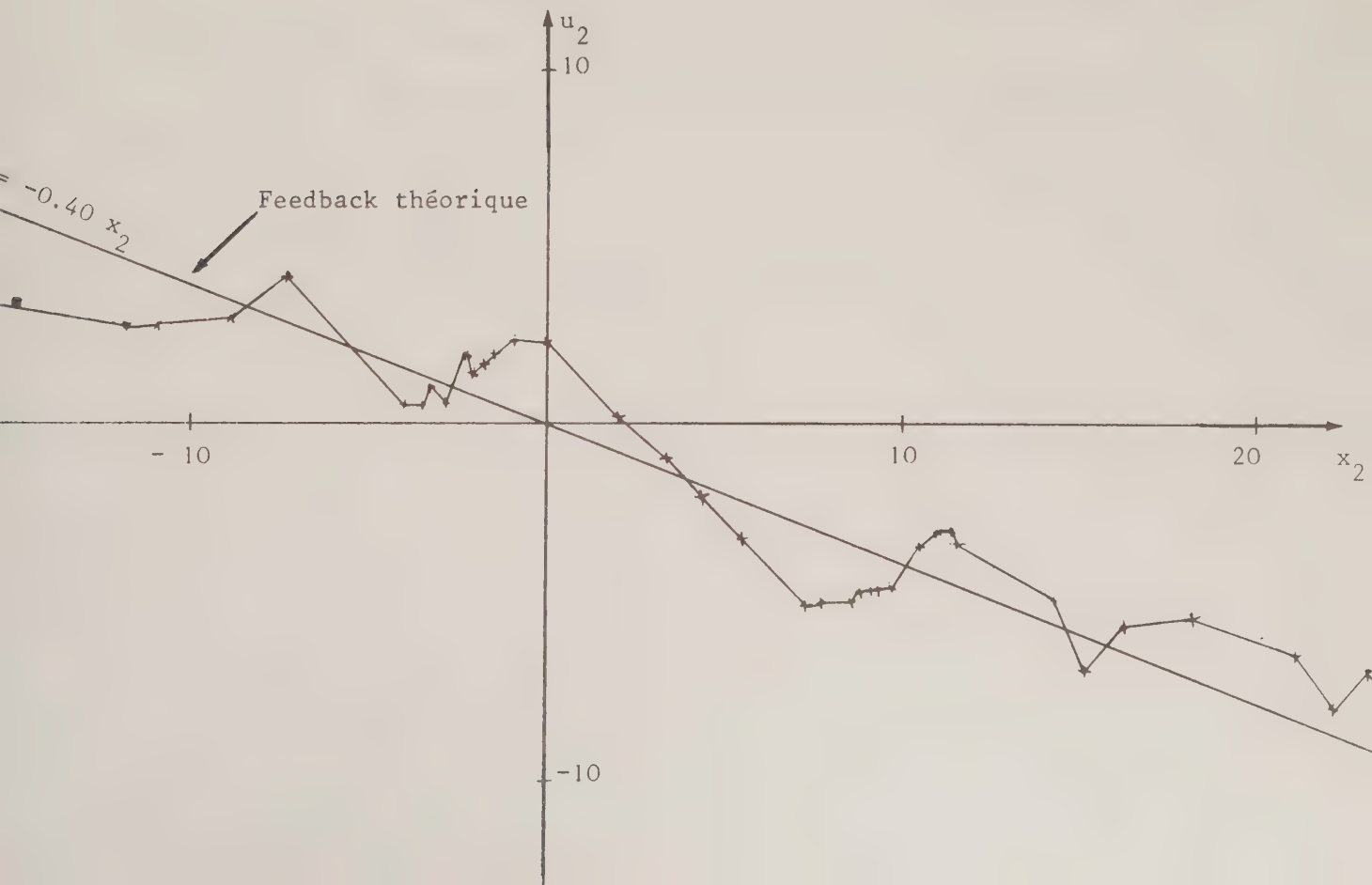


5.4.3. P_r 50 réalisations

5.4.3.1. $u_1 = f(x_1)$



5.4.3.2. $u_2 = f(x_2)$



6 - Conclusion.

Cet exemple montre que :

a) la commande initiale.

est bonne pour O_r, P_r, R_r dés que l'on tire les réalisations du bruit par couples symétriques.

b) les autres commandes

-sont bonnes pour O_r si on désire les commandes open loop optimales

-sont difficilement utilisables pour P_r et R_r .

c) le coût

-de O_r converge relativement vite vers le coût open loop lorsqu'on augmente le nombre de réalisations,

-de P_r converge plus lentement vers le coût feedback optimal. On note l'influence de la classe de stratégie sur laquelle on minimise même si celle ci contient la stratégie Feedback optimale.

-de R_r donne une mauvaise approximation du coût feedback optimal (beaucoup trop faible).

d) temps de calcul

trés favorable aux problèmes O_r et R_r

§-2 EXEMPLE II

SYSTEME PRODUCTION-STOCK D'UN PRODUIT

1. Le problème $P(\mu)$

(1) $$x_{i+1} = x_i + u_i - d_i \qquad i=1,\ldots 9$$

x_1 donné

d_i lois uniformes sur $(0,3)$ indépendants deux à deux

(2) $$0 \leqslant u_i \leqslant 5 \qquad i=1,\ldots 9$$

(3) $$P\left(0 \leqslant x_i \leqslant 10\right) = 1 \qquad i=2,\ldots 9$$

(4) $$\text{Minimiser} \sum_{i=1}^{9} \left(u_i + (u_i - u_{i-1})^2 - \frac{x_i}{2} - 0{,}5 + \frac{1}{2}(9x_i^2 + 2x_i + 1)^{\frac{1}{2}}\right)$$

où :

- x_i sera le niveau de stock à l'instant i ,
- u_i sera le niveau de production entre les instants i et i+1,
- d_i sera la demande entre les instants i et i+1.

(2) représente des contraintes sur le système de production.

(3) représente une contrainte sur la quantité de stockage ($x_i \leqslant 10$) et la volonté de répondre à la demande ($x_i \geqslant 0$) (Ces deux contraintes devant être réalisées presque surement).

Le critère à minimiser se compose de trois coûts :

- Coût de production linéaire u_i ,
- Un coût de stockage $-\frac{x_i}{2} - 0,5 + \frac{1}{2}(9\, x_i^2 + 2\, x_{i+1})^{\frac{1}{2}}$,
- Un coût de variation de niveau de production $(u_i - u_{i-1})^2$.

Remarque 1

$-\frac{x_i}{2} - 0,5 + \frac{1}{2}\,(9x_i^2 + 2x_i + 1)^{\frac{1}{2}}$ est une hyperbole tangente à l'origine, très proche des deux asymptotes :

$$y = x - \frac{1}{2},$$

$$y = -2x - \frac{2}{3}\ .$$

Elle sert à approcher la fonction coût suivante :

$$\begin{cases} y = x & \text{si} \quad x \geqslant 0 \\ y = -2x & \text{si} \quad x \leqslant 0 \end{cases}$$

en la rendant différentiable.

Remarque 2

Ce problème trés simple présente les principales difficultés que nous rencontrons pour les systèmes production-stock :

- Avenir aléatoire,
- Contrainte sur le stock et le niveau de production.

Il comporte cependant des simplifications :

- Le critère est convexe (ce n'est pas toujours le cas pour des problèmes pratiques),
- Les contraintes sur l'état sont souvent en probabilité (la résolution du problème avec de telles contraintes aurait été possible mais trop couteuse (chII. §4)).

Remarque 3

Le critère fait intervenir pour le coût entre i et $i+1$ la valeur de u_i et de u_{i-1}. Ceci est inacceptable pour nos algorithmes, aussi allons-nous écrire le problème équivalent suivant :

$$
\left\{
\begin{array}{ll}
\left\{
\begin{array}{ll}
x^1_{i+1} = x^1_i + u_i - d_i & x^1_o \text{ donné} \\
x^2_{i+1} = u_i & x^2_o \text{ donné} = u_{-1} \quad i=1,\dots 9 \\
x^3_{i+1} = x^2_i & x^3_o \text{ donné} = u_{-1}
\end{array}
\right. & \text{(1')} \\
0 \leq u_i \leq 5 & \\
P\left(0 \leq x^1_i \leq 10 \right) = 1 & \\
\underset{U}{\text{Min}} \sum_{i=1}^{9} \left(u_i + (x^2_i - x^3_i)^2 + \frac{1}{2}(9(x^1_i)^2 + 2x^1_i + 1)^{\frac{1}{2}} - 0{,}5 - \frac{x^1_i}{2} \right) &
\end{array}
\right.
$$

Les équations d'évolution (1') font apparaître un état de dimension 3. Il aurait été possible de réduire la dimension de l'état à 2 en choisissant comme contrôle la variation de production. Nous ne l'avons pas fait afin

de mieux montrer les limites de la programmation dynamique classique.

Remarque 4

La contrainte (3) peut s'écrire également

$$(5) \qquad P\left(0 \leqslant x_{i-1} + u_{i-1} - d_{i-1} \leqslant 10\right) = 1 \qquad i=1,\ldots,9$$

Appelons d_{min} (resp d_{max}) la borne inférieure (resp supérieure) du support de d .

(5) peut s'écrire alors

$$(6) \qquad \begin{cases} P\left(0 \leqslant x_{i-1} + u_{i-1} - d_{max}\right) = 1 \\ \\ P\left(10 \geqslant x_{i-1} + u_{i-1} - d_{min}\right) = 1 \end{cases} \qquad i=1,\ldots,9$$

Le problème (1')(2)(3)(4) sera noté $P^1(\mu)$

Le problème (1')(2)(6)(4) sera noté $P^2(\mu)$

$P^1(\mu)$ et $P^2(\mu)$ sont évidemment équivalents.

2. Résultats obtenus par la programmation dynamique classique (PDC)

(chII;§3.1;B)

- Chaque composante de l'état est discrétisée en 6 points équidistants

$$x_i^1 \in \{2k \ ; \ k \in 0,\ldots,5\}$$

$$i=1,\ldots,9$$

$$x_i^2 \ , \ x_i^3 \in \{0,1,\ldots,5\}$$

- Le bruit a été discrétisé de deux façons :

$$(7) \qquad e_i^j = 0.15 + 0.3(j-1) \qquad j \in \{1,\ldots,10\}$$

$$(7') \qquad e_i^j = 0.3(j-1) \qquad j \in \{1,\ldots,11\}$$

- Le théorème 1 du chII § 3 s'appliquant, la commande optimale a été obtenue par une localisation du type dichotomie utilisant les nombres de FIBONACCI(BELLMAN (14)).

2.1. Discrétisation du bruit (7)

- Temps de calcul : CII 10 070 12mn pour l'obtention de toute la stratégie.

- Tableau donnant la valeur de la commande initiale optimale en fonction de l'état initial $(x_o, u-1)$

x_o \ u_{-1}	0	2	4
0	2,85	2,85	2,99
2	0,99	1,43	2,01
4	0,01	0,99	1,97
6	0,01	0,44	1,00
8	0,00	0,00	1,00

(PDC)

- Tableau donnant la valeur du coût optimal en fonction de l'état initial (x_o, u_{-1})

x_o \ u_{-1}	0	2	4
0	41,7	34,3	34,8
2	34,2	33,9	38,2
4	34,5	36,1	43,0
6	36,8	40,6	48,7
8	41,8	45,8	55,3

(PDC)

2.2. Discrétisation du bruit (7')

- Temps de calcul : CII 10 070
- Tableau donnant la valeur de la commande initiale optimale en fonction de l'état initial (x_o, u_{-1})

x_o \ u_{-1}	0	2	4
0	3,00	3,00	3,00
2	1,00	1,58	2,00
4	0,18	1,01	1,97
6	0,00	0,75	1,09
8	0,00	0,00	1,00

(PDC)

- Tableau donnant la valeur du coût optimal en fonction de l'état initial (x_o, u_{-1})

x_o \ u_{-1}	0	2	4
0	44,9	36,8	36,8
2	36,8	36,2	40,3
4	37,1	38,2	45,0
6	39,1	42,8	50,9
8	43,9	47,8	57,2

(PDC)

Remarque 5

Les discrétisations du bruit (7) et (7') conduisent à des résultats qui différent sensiblement :

- environ 6% pour les coûts optimaux,

Ces différences permettent de se faire une idée de la qualité des résultats obtenus par la programmation dynamique classique.

Nous allons, maintenant, pouvoir comparer ces résultats avec ceux obtenus par les méthodes de gradients et d'UZAWA appliqués aux problèmes O_r^1, R_r^1, P_r^1, O_r^2, R_r^2, P_r^2 (l'indice supérieur 1 ou 2 correspond aux problèmes $P^1(\mu)$ ou $P^2(\mu)$

définis au 1)

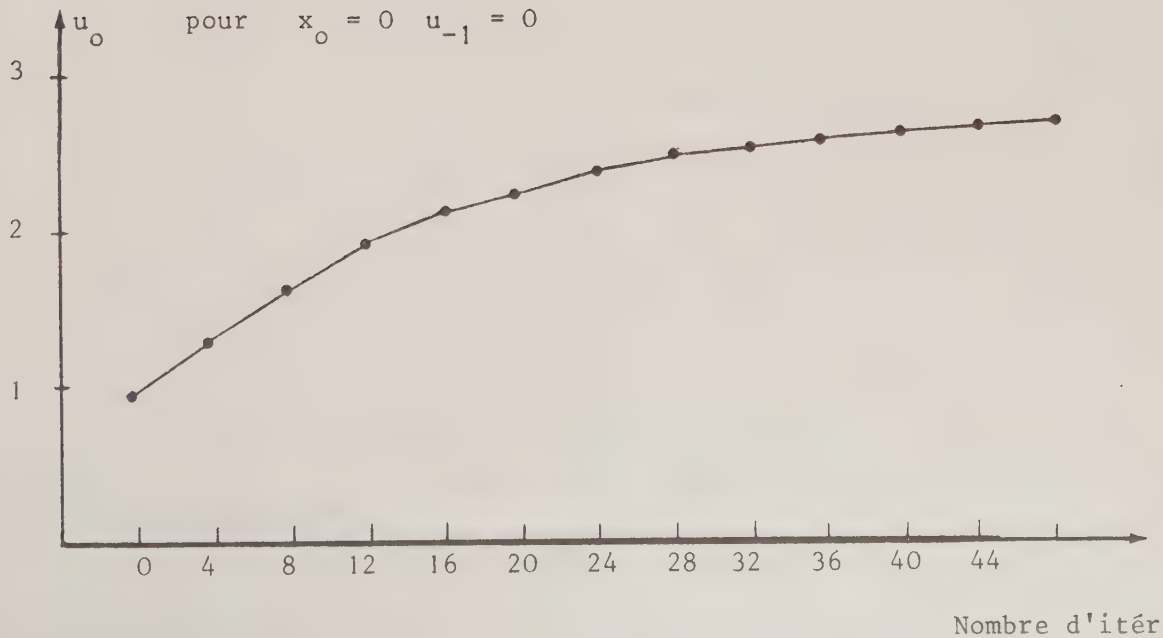
3. Résolution des problèmes open loop 0_r^1 et 0_r^2 .

3.1. Résultats obtenus par l'algorithme d'UZAWA appliqué au problème 0_r^1 .

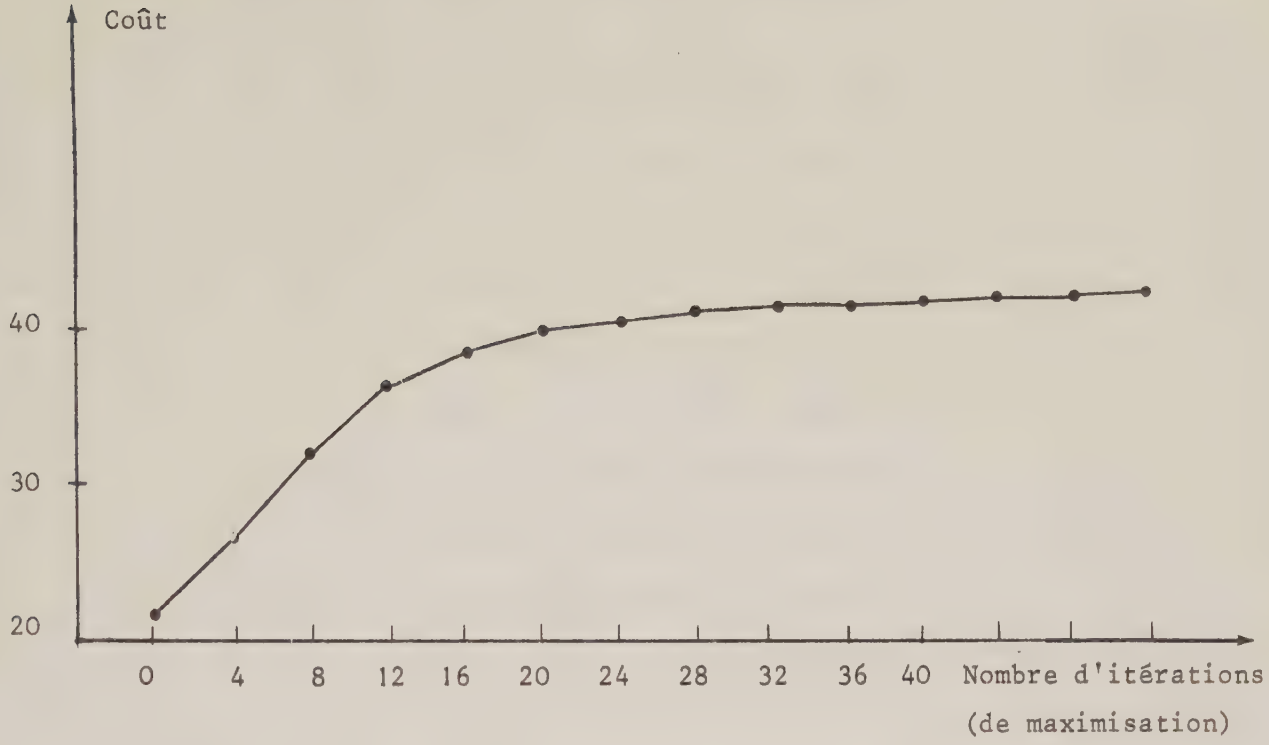
(Seules les contraintes du type (3) sont dualisées)

- Nombre de réalisations : r=10 (0_{10}^1)
- Temps machine : CII 10 070 : en moyenne 1 mn (pour une condition initiale donnée (x_o,u_{-1})). Ce temps peut varier selon la proximité à la solution optimale, de la commande servant à l'initialisation de l'algorithme.
- Convergence de u_o pour x_o=0 et u_{-1} = 0 en fonction du nombre d'itérations :

(Une itération correspond à une étape de maximisation dans la recherche du point-selle).



- Convergence du coût en fonction du nombre d'itérations (pour $x_o=0$ et $u_{-1} = 0$)



- Tableau donnant la valeur de la commande initiale en fonction de l'état initial (x_o, u_{-1}).

x_o \ u_{-1}	0	2	4
0	2,75	2,87	2,89
2	0,92	1,25	2,21
4	0,0	0,88	1,85
6	0,0	0,61	1,59
8	0,0	0,37	1,34

(0^1_{10})

- Tableau donnant la valeur du coût optimal en fonction de l'état initial (x_o, u_{-1}).

x_o \ u_{-1}	0	2	4
0	42,75	34,92	34,93
2	33,58	33,40	38,60
4	35,02	37,13	43,43
6	38,42	41,43	48,80
8	42,45	46,10	54,54

(0^1_{10})

Remarque 1

Il n'existe pas de stratégies open loop satisfaisant la contrainte (3).

Considérons en effet les deux réalisations suivantes de la demande

$$d_i^1 = 3 \qquad i=1,\ldots,9$$

$$d_i^2 = 0 \qquad i=1,\ldots,9$$

Les états à l'instant 9 seront alors

$$x_9^1 = x_1 + \sum_{i=1}^{8} u_i - 24$$

$$x_9^2 = x_1 + \sum_{i=1}^{8} u_i$$

Le support de la variable $x_9(u)$ ou $u=(u_1,\ldots,u_8)$ est le segment

$$\left(x_1 + \sum_{i=1}^{8} u_i - 24,\ x_1 + \sum_{i=1}^{8} u_i \right)$$

qui contredit (3) quelque soit la stratégie open loop choisie.

Remarque 2

Malgré la remarque 1, on a obtenu pour O_r^1 une solution raisonnable (comparer avec la programmation dynamique classique).

Ce résultat s'explique par la probabilité faible des réalisations extrêmes. Les 10 réalisations tirées étaient compatibles avec la contrainte (3).

3.2. Le problème O_r^2 n'a pas de solution du fait de la remarque 1.

4. Résolution des problèmes relaxés R_r^1 et R_r^2.

4.1. Algorithme d'UZAWA appliqué au problème R_r^1.

(- Seules les contraintes correspondant à (3) sont dualisées.

- Ici la contrainte feedback (4' chII) est intégrée dans l'équation d'évolution donc n'apparaît pas comme contrainte sur l'état)

- Nombre de réalisations : r = 10
- Temps machine CII 10 070 : 20s (pour une condition initiale (x_o, u_{-1}) déterminée.
- Commande initiale et coût optimaux en fonction de l'état initial

Identique au troisième chiffre significatif prés aux tableaux de 4.2.

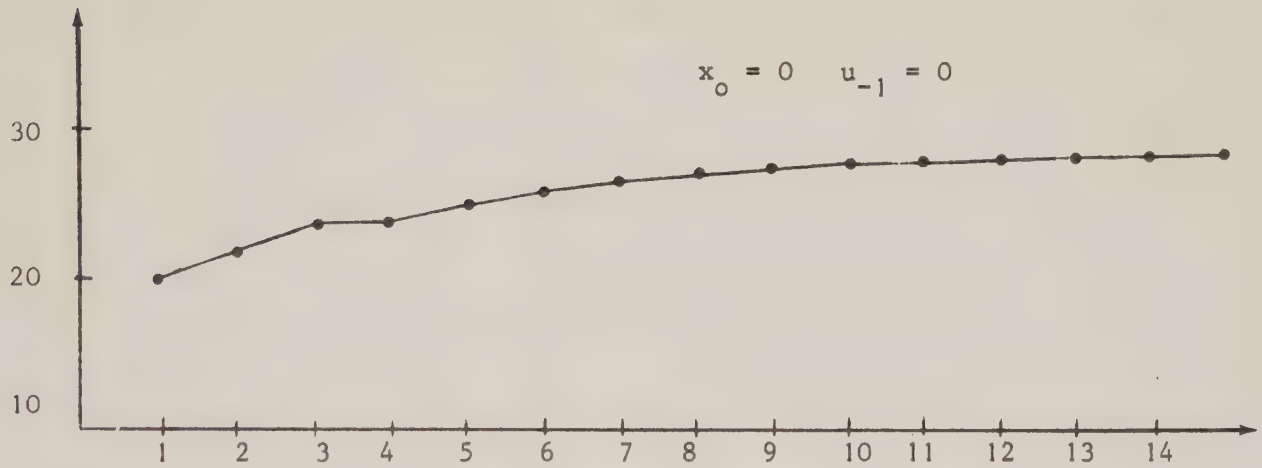
4.2. Algorithme d'UZAWA utilisé pour résoudre le problème de décomposition (chII § 2.2.2.) associé au problème R_r^1.

(Les contraintes correspondant à (3) pour chaque problème local sont dualisées).

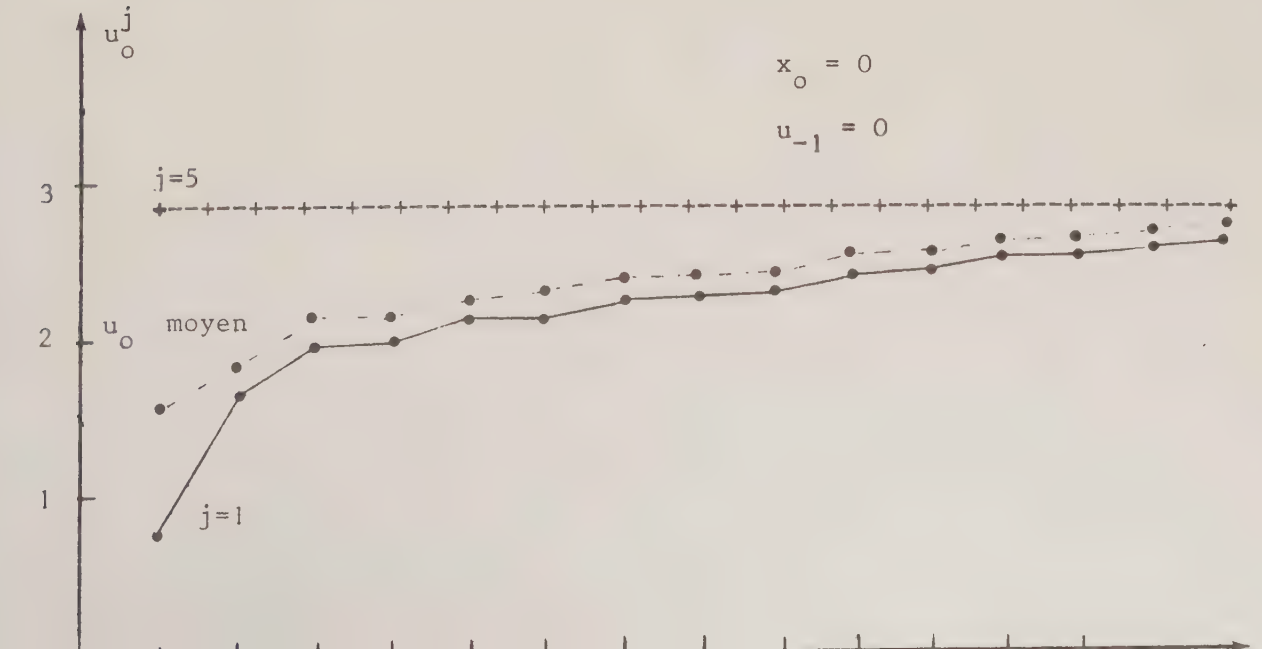
- Nombre de réalisations : r = 10 (R_{10}^{1})

- Temps machine : CII 10 070 : 1mn (pour la résolution du problème avec une condition initiale (x_o,u_{-1}) déterminée).

- Convergence du coût en fonction du nombre d'itérations de coordination :



- Coordination des u_o^j au point initial $x_o = 0$, $u_{-1}=0$, en fonction du nombre d'itérations de coordination :



- Tableau donnant la valeur de la commande initiale optimale en fonction de l'état initial (x_o, u_{-1})

x_o \ u_{-1}	0	2	4
0	2,80	2,80	2,80
2	0,80	0,98	1,74
4	0	0,52	1,38
6	0	0,19	1,08
8	0	0	0,84

(R^1_{10})

- Tableau donnant la valeur du coût optimal en fonction de l'état initial

x_o \ u_{-1}	0	2	4
0	28,1	20,2	20,0
2	16,4	16,6	23,2
4	17,5	20,8	29,1
6	22,7	26,7	36,1
8	30,2	34,2	44,6

(R^1_{10})

4.3. Algorithme de pénalisation (ch II § 2.1) appliqué au problème R_r^2

(des contraintes correspondant à (3) sont pénalisées).

- Temps de calcul : CII 10 070 (2mn par condition initiale x_o, u_{-1})
- Nombre de réalisations : r=10
- Tableau donnant la valeur de la commande initiale optimale en fonction de l'état initial (x_o, u_{-1})

x_o \ u_{-1}	0	2	4
0	2,96	2,98	2,98
2	1,00	1,42	2,06
4	0,19	0,85	1,63
6	0,00	0,41	1,25
8	0,00	0,1	0,96

(R_{10}^2)

- Tableau donnant la valeur du coût optimal en fonction de l'état initial (x_o, u_{-1})

x_o \ u_{-1}	0	2	4
0	38,4	30,4	30,3
2	29,0	28,3	33,4
4	29,4	31,5	38,6
6	32,4	36,0	44,8
8	38,4	42,4	52,5

(R_{10}^2)

Remarque 1

- La différence de temps de calcul entre (4.1) et (4.2) l'explique. Chaque résolution d'un problème local dans (4.2. nécessite la recherche d'un point-selle. (Ces derniers contiennent des contraintes sur l'état (3)).
- La différence de temps de calcul entre (4.1) et (4.3) s'explique par la différence du nombre de contraintes actives entre les deux problèmes.

Remarque 2

Nous avons essayé l'algorithme d'UZAWA sur le problème R_r^2 . Il était moins performant que l'algorithme de gradient sur le problème pénalisé.

Remarque 3

La solution optimale du problème R_r^2 est plus proche des résultats de la P.D.C. (programmation dynamique classique) que la solution de R_r^1 .
En effet la classe de stratégies admissibles pour le problème R_r^2 est plus petite que celle de R_r^1 .

5. Résolution du problème Feedback P_r^2.

Les problèmes P_r^2 étant de meilleures approximations de $P(\mu)$ que P_r^1 nous étudierons seulement P_r^2 .
La non convexité du problème P_r^2 (ch II. § 1.2) nous permettra d'obtenir seulement un minimum local.
Nous utiliserons exclusivement des méthodes de pénalisation pour résoudre P_r^2 . En effet la non convexité de P_r^2 pose des difficultés pour les méthodes duales.
Nous étudierons l'influence de la classe de stratégies $\mathcal{Y}_k$ (chII §1.1). Nous optimiserons P_r^2 dans $\mathcal{Y}_1$ et $\mathcal{Y}_{100}$.

Nous étudierons également l'influence du nombre de réalisations. Nous résoudrons P_r^2 pour 10 et 20 réalisations.

- Classe de stratégie dans laquelle on minimise : $\mathcal{S}_1$
- nombre de réalisations : 10
- Temps de calcul : CII 10 070 : $\sim$2mn
- Tableau donnant la valeur de la commande initiale optimale en fonction de l'état initial (x_o, u_{-1})

x_o \ u_{-1}	0	2	4
0	2,94	2,98	2,98
2	1,00	1,42	2,08
4	0,20	0,86	1,62
6	0,00	0,43	1,23
8	0,00	0,1	0,97

(P_{10}^2) dans $\mathcal{S}_1$

- Tableau donnant la valeur du coût optimal en fonction de l'état initial (x_o, u_{-1})

x_o \ u_{-1}	0	2	4
0	38,7	30,7	30,5
2	29,1	28,5	33,5
4	29,6	31,6	38,7
6	32,5	36,2	45,1
8	38,9	42,7	52,7

(P^2_{IO}) dans φ_1

- Classe de stratégie dans laquelle on minimise : $\mathcal{S}_1$
- Nombre de réalisations : 20
- Temps de calcul : CII 10 070 : $\sim$ 4 mn
- Tableau donnant la valeur de la commande initiale optimale en fonction de l'état initial (x_o, u_{-1})

x_o \ u_{-1}	0	2	4
0	2,89	2,97	2,98
2	1,00	1,35	2,09
4	0,24	0,75	1,66
6	0,0	0,35	1,30
8	0,0	0,12	0,99

(P^2_{20}) dans $\mathcal{S}_1$

- Tableau donnant la valeur du coût optimal en fonction de l'état initial (x_o, u_{-1})

x_o \ u_{-1}	0	2	4
0	38,7	30,9	30,6
2	29,2	28,6	34,2
4	29,8	32,0	39,0
6	33,0	36,5	45,3
8	38,8	42,7	52,8

(P_{20}^2) dans $\mathcal{S}_1$

- Classe de stratégies dans laquelle on minimise : $\mathcal{S}_{100}$
- Nombre de réalisations : 10
- Temps de calcul : CII 10 070 : ~2mn
- Tableau donnant la valeur de la commande initiale optimale en fonction de l'état initial (x_o, u_{-1})

x_o \ u_{-1}	0	2	4
0	2,96	2,98	2,98
2	1,00	1,42	2,06
4	0,19	0,85	1,63
6	0,00	0,41	1,25
8	0,00	0,10	0,96

(P_{10}^2) dans $\mathcal{S}_{100}$

- Tableau donnant la valeur du coût optimal en fonction de l'état initial (x_o, u_{-1})

x_o \ u_{-1}	0	2	4
0	38,4	30,4	30,3
2	29,0	28,3	33,4
4	29,4	31,5	38,6
6	32,4	36,0	44,8
8	38,8	42,5	52,5

(P_{10}^2) dans $\mathcal{Y}_{100}$

Remarque

Nous constatons sur ce problème la faible influence du nombre de réalisations et de la classe de stratégies sur laquelle on minimise.

6. Conclusion.

- Sur ce problème apparaît la nécessité de chercher la solution du problème $P(\mu)$ dans une autre classe de stratégies que l'open loop.

- La résolution du problème R_r^2 semble suffisante pour ce problème. Les problèmes P_r^2 ne donnent pas de solutions meilleures.

- Pour les problèmes P_r^2 nous voyons apparaître la faible influence :

  - de la classe de stratégies sur laquelle on minimise,

  - du nombre de réalisations.

- La comparaison des solutions du problème P_r^2 ou R_r^2 avec celles obtenues par la P.D.C. (2.1) fait apparaître des variations de l'ordre de 10%. Les temps de calcul étant trés nettement à l'avantage des problèmes P_r^2 ou R_r^2 dans un rapport de 1 à 6.

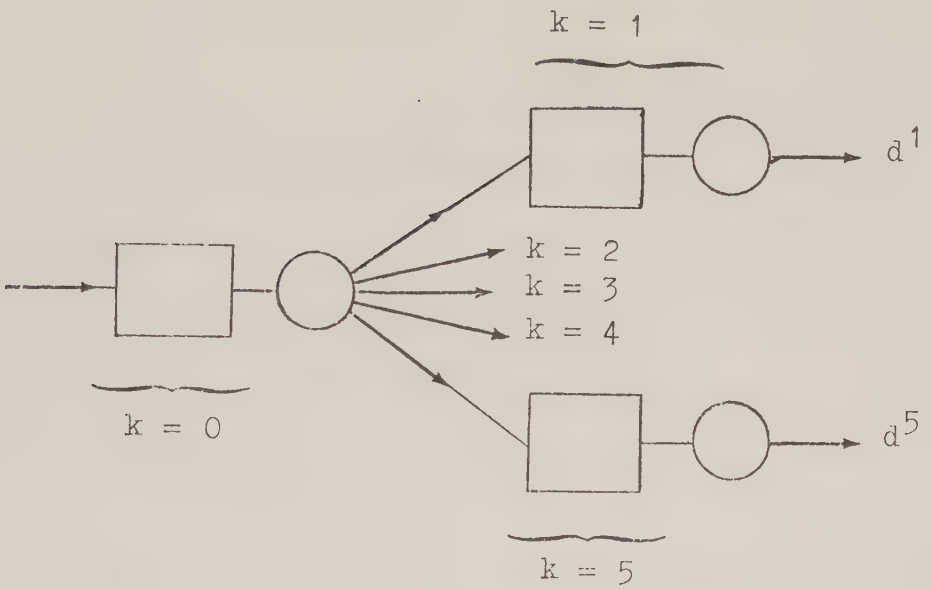
§ - 3

EXEMPLE III

SYSTEME PRODUCTION-STOCK UNE RESSOURCE 5 PRODUITS

1. Le problème $P(\mu)$

Le système peut être représenté par le diagramme



$\boxed{\cdot}$ désigne un système de production régi par l'équation

(1) $$p_{i+1}^k = p_i^k + u_i^k \qquad i=0,1,\ldots,8 \quad \text{pour} \quad k=0,1,\ldots 5$$

- p_i^k désigne le niveau de production du produit k entre les instants i et $i+1$.
- u_i^k désigne la variation de production du produit k à l'instant i .

( ) représente un stock régi par l'équation

(2) $$s_{i+1}^k = s_i^k + p_i^k - d_i^k \qquad i = 0,\ldots,8 \quad \text{pour} \quad k=0,1,\ldots 5$$

- s_i^k désigne le niveau de stock du produit k à l'instant i.
- d_i^k désigne la demande entre les instants i et $i+1$.

Remarque 1

- de produit 0 sera appelé ressource
- La demande de la ressource sera :

(3) $$d_i^o = \sum_{k=1}^{5} p_i^k \qquad \text{pour} \quad i=0,\ldots,8$$

Les demandes d_k^i $k=1,\ldots,5$; $i=0,\ldots,8$ seront des variables aléatoires indépendantes, de loi de probabilité égale au produit de convolution de trois lois uniformes sur $(0,0.1)$

De plus le système comportera les contraintes suivantes

- Sur les niveaux de production

(4) $$0 \leqslant p_i^k \leqslant 2 \qquad k=0,\ldots,5 \ ; \ i=0,\ldots,8$$

- Sur le stock

(5) $$s_i^k \geqslant 0 \quad \text{p.s.} \qquad k=0,\ldots,5 \ ; \ i=1,\ldots,8$$

Le critère à minimiser sera :

$$E \left\{ \sum_{i=0}^{8} \left(p_i + 0{,}2\, s_i + u_i^2 \right) \right\} \tag{6}$$

Pour une étude systématique de ce modèle dans le cas déterministe : LEMARECHAL, NEPOMIASTCHY, NISSEN (15), POTIER (16).

Remarque 2

Comme dans l'exemple 2 (§2.1) nous remplacerons la contrainte (5) par la contrainte équivalente suivante

$$s_{i-1}^k + p_{i-1}^k - d_{max}^k \geqslant 0 \quad \text{p.s.} \quad k=1,\ldots,5 \; ; \; i=1,\ldots,8 \tag{7}$$

où d_{max}^k désigne la demande maximum du produit k .

Nous donnerons les résultats uniquement pour les problèmes P_r .

Nous chercherons la solution dans la classe $\mathcal{G}_1$ (ch II §1.1)

Remarque 3

Les problèmes relaxés R_r donnent les mêmes résultats que les problèmes P_r (les contraintes feedback (ch II (4)) sont presque toujours inactives). Les courbes qui suivent sont les résultats de l'utilisation de l'algorithme de gradient appliqué au problème P_r pénalisé, utilisé en temps réel. D'une façon plus explicite cela consiste en :

(I) Résolution du problème P_r ,

(II) Simulation de la demande $d_o = (d_o^k \quad k=0,\ldots,5)$

(III) Intégration du système (1)(2) en utilisant :

- La commande initiale obtenue par (I),
- La demande obtenue par (II).

Nous obtenons ainsi un nouvel état

$$(p_1,s_1) = (p_1^k,s_1^k \; ; \quad k=0,\ldots,5)$$

(IV) Construire un nouveau problème P_r avec

- l'état initial (p_1,s_1)
- le temps variant de 0 à n-1 (où n est le nombre d'étapes dans le temps dans (I)).

(V) Retourner en (I) jusqu'à ce que n=1.

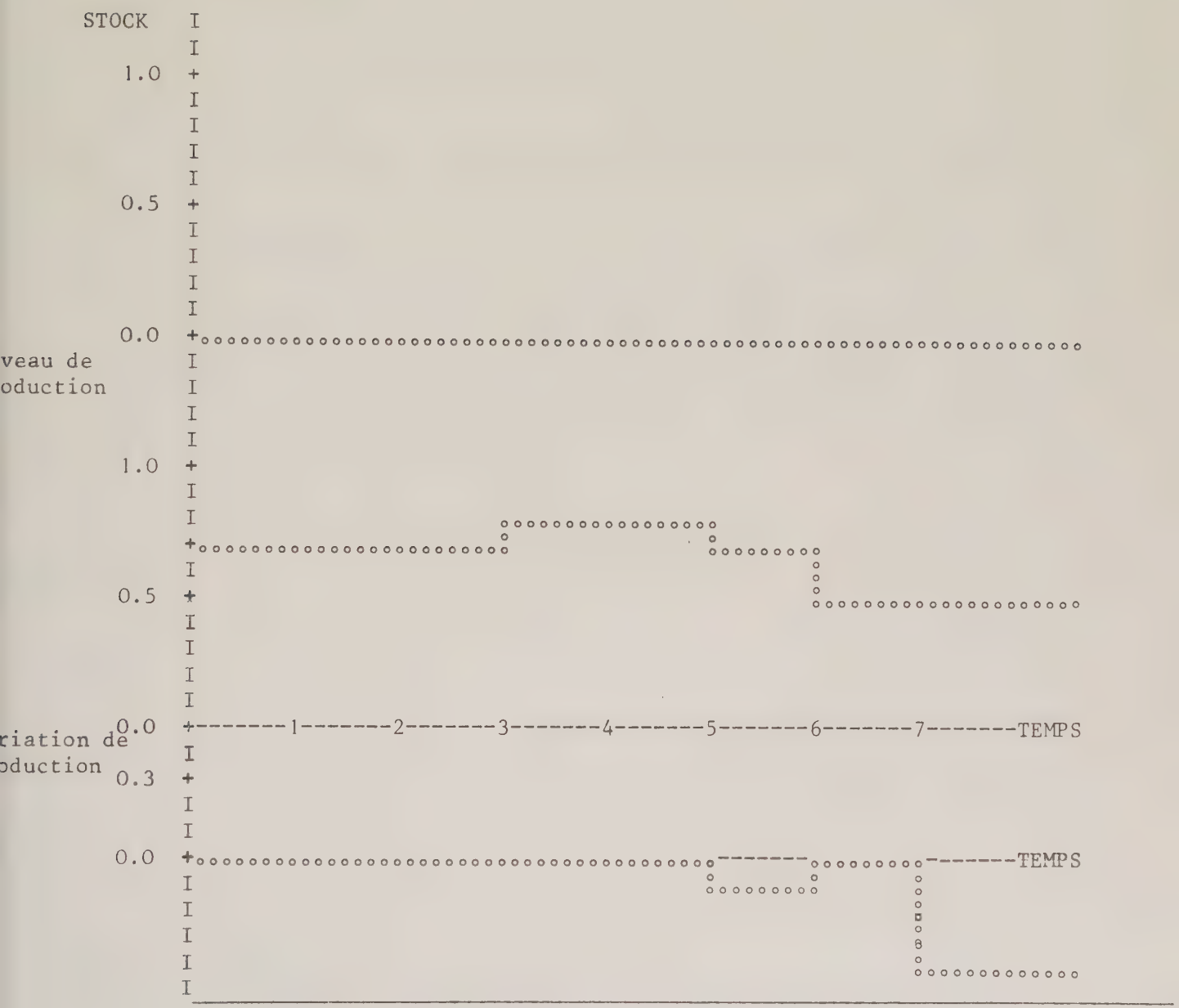
Remarque 4

Dans les problèmes P_r les contraintes pénalisées sont (7) et la contrainte feedback (chII §1 (4)).

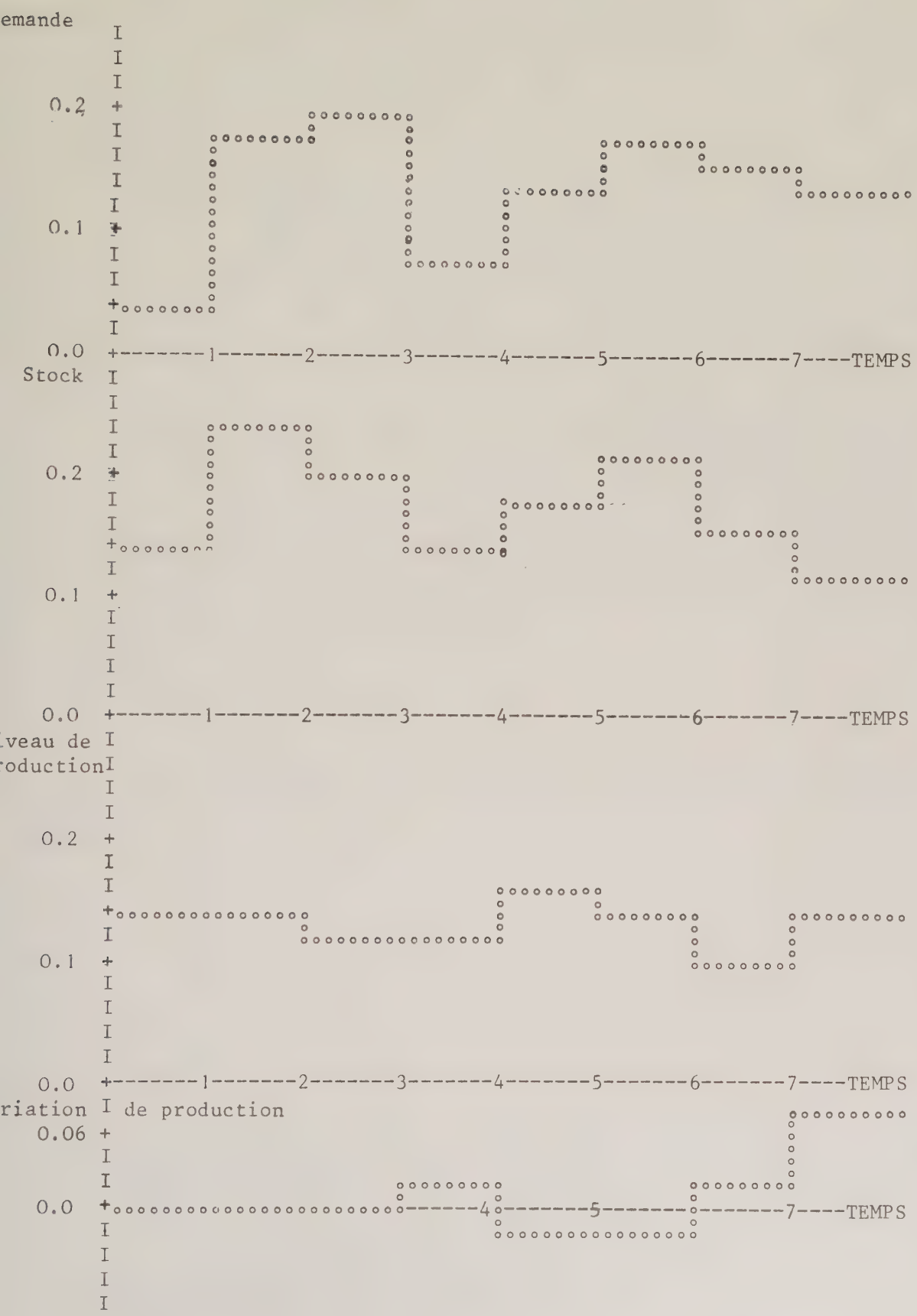
2. Résultats de l'utilisation de l'algorithme P_r en temps réel.

2.1.

RESSOURCE

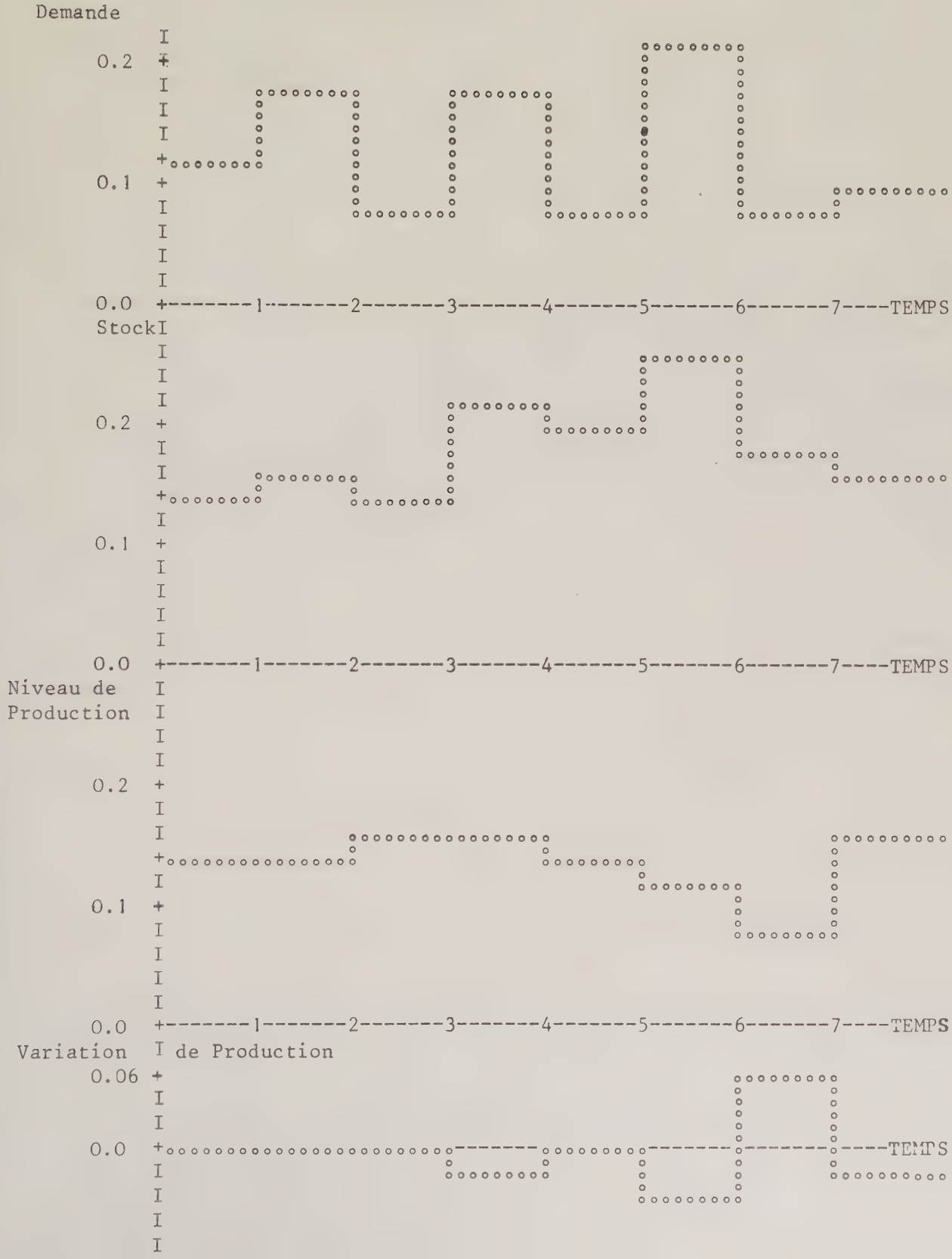


2.2 PRODUIT 1


emande
0.2
0.1
0.0
1
2
3
4
5
6
7
TEMPS
Stock
0.2
0.1
0.0
TEMPS
veau de
oduction
0.2
0.1
0.0
TEMPS
riation
de production
0.06
0.0
TEMPS

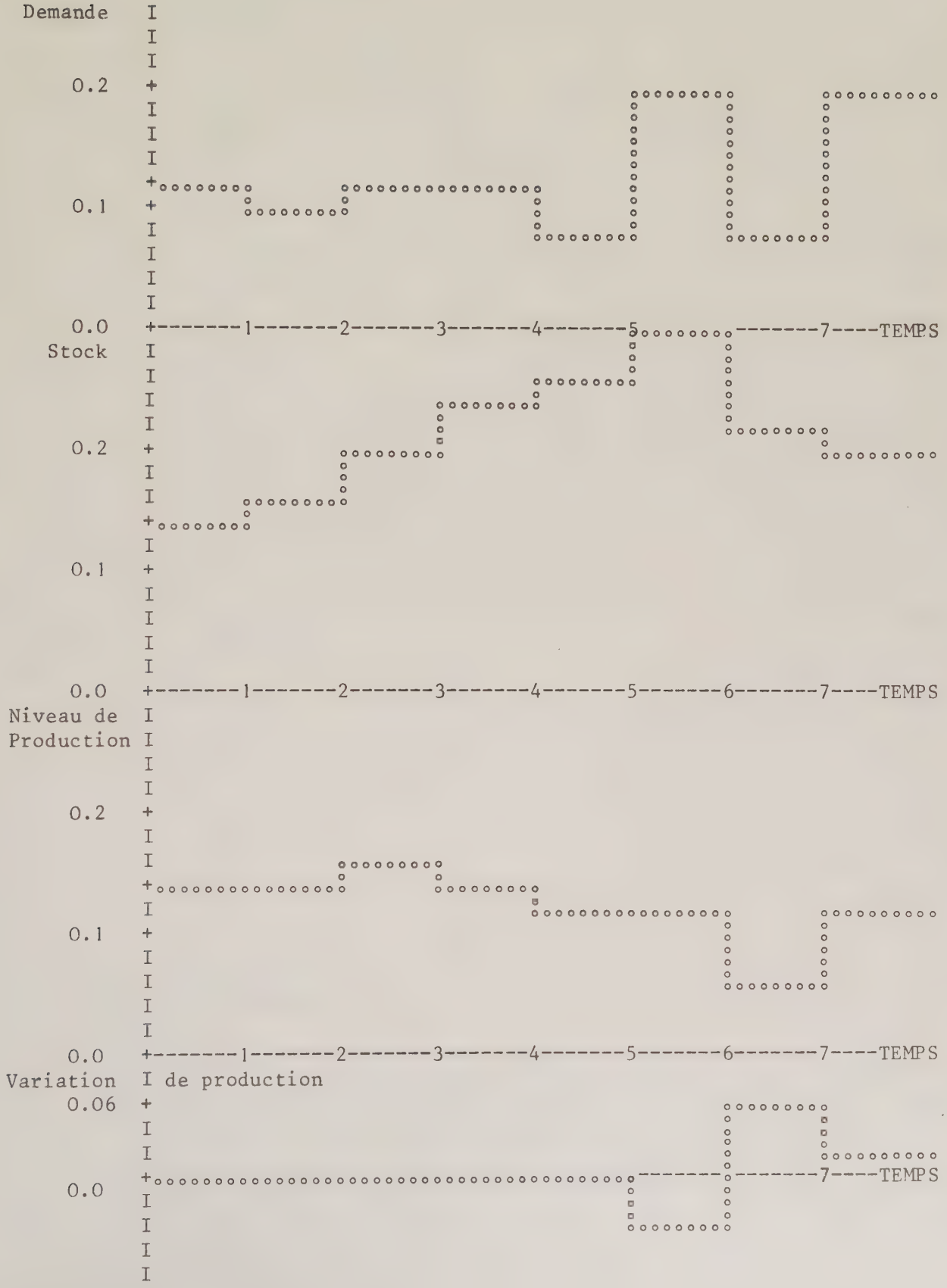
2.3.

PRODUIT 2


Demande
0.2
0.1
0.0
1
2
3
4
5
6
7
TEMPS
Stock
0.2
0.1
0.0
TEMPS
Niveau de Production
0.2
0.1
0.0
TEMPS
Variation de Production
0.06
0.0
TEMPS

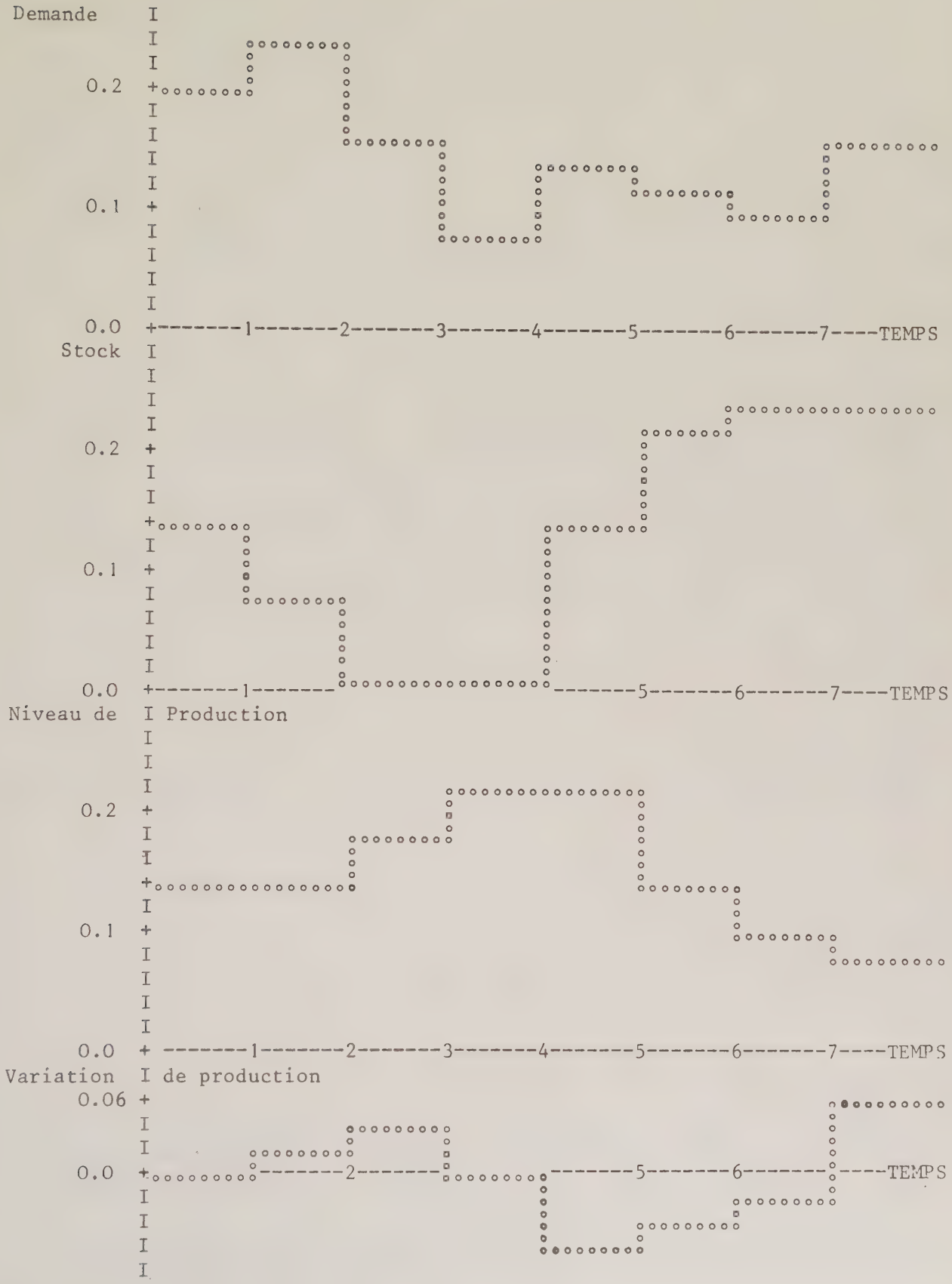
2.4

PRODUIT 3


Demande
0.2
0.1
0.0
1
2
3
4
5
6
7
TEMPS
Stock
0.2
0.1
0.0
Niveau de Production
0.2
0.1
0.0
Variation de production
0.06
0.0

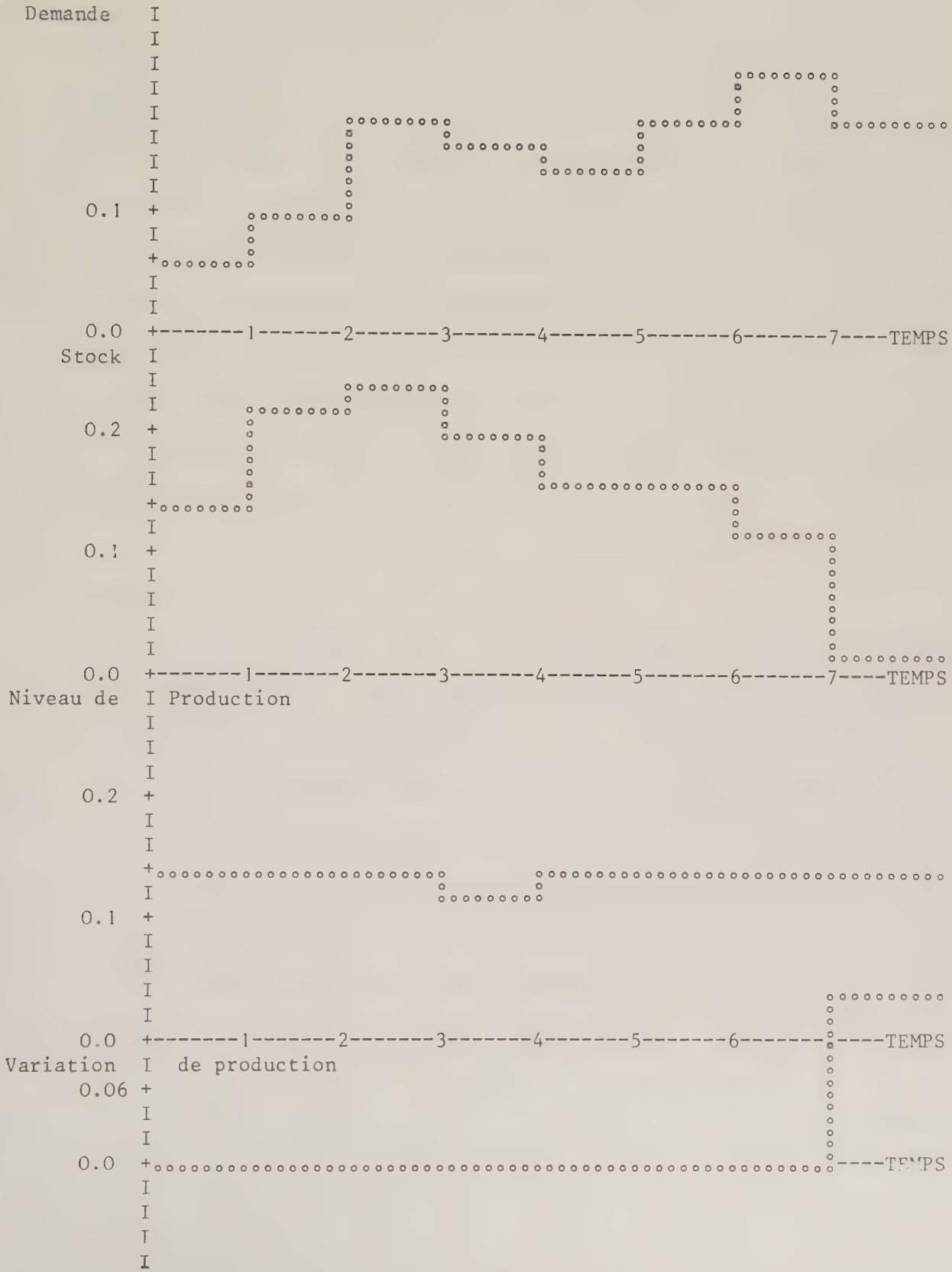
2.5.

PRODUIT 4


Demande
0.2
0.1
0.0
1
2
3
4
5
6
7
TEMPS
Stock
0.2
0.1
0.0
1
5
6
7
TEMPS
Niveau de Production
0.2
0.1
0.0
1
2
3
4
5
6
7
TEMPS
Variation de production
0.06
0.0
2
5
6
TEMPS

2.6.

PRODUIT 5


Demande
0.1
0.0
1
2
3
4
5
6
7
TEMPS
Stock
0.2
0.1
0.0
TEMPS
Niveau de Production
0.2
0.1
0.0
TEMPS
Variation de production
0.06
0.0
TEMPS

Remarque 5

Pour la ressource nous n'avons pas donné la courbe de la demande. Elle est égale à la somme des niveaux de production des autres produits (3).
Le nombre de réalisations utilisé pour la construction de P_r est 10.
Le temps de calcul pour la résolution de P_r avec 9 étapes dans le temps environ 3mn CII 10 070.
Le coût optimum obtenu par la résolution de P_r est 16,6.
Le coût de la trajectoire optimale obtenue en appliquant l'algorithme en temps réel est 14,8.

Remarque 6

La résolution du problème P_r associé à cet exemple a permis l'obtention de la solution en un temps raisonnable alors qu'il aurait été impossible d'appliquer la P.D.C. directement.
D'autre part cet exemple ne peut pas être traité par la méthode de KALMAN car il possède des contraintes sur l'état (4) et (5). Il est donc difficile de se faire une idée de la qualité de la solution obtenue sur cet exemple. Il montre cependant qu'il est possible de traiter (par les méthodes exposés au ch II). Une classe plus grande de problèmes que par les méthodes usuelles.

§ - 4 CONCLUSION DES EXEMPLES NUMERIQUES

4. Conclusion des exemple numériques.

- Les méthodes O_r, P_r, R_r permettent de résoudre une classe plus grande de problèmes avec les méthodes classqiues (KALMAN, programmation dynamique) (ex 3).
- Ils sont acceptables pour une utilisation en temps réels car la commande initiale est bonne.

- Une bonne approximation du coût peut nécessiter dans certains cas (ex 1) la résolution de P_r avec r grand ; dans d'autre cas (ex 2) R_r et P_r fournissent dé bonnes approximations du coût optimal feedback pour r beaucoup plus petit.
- Si l'on désire une stratégie open loop alors O_r fournit la stratégie optimale complète et non plus seulement la commande initiale.

BIBLIOGRAPHIE

(1) BERGE C. Espaces topologiques fonctions multivoques
Dunod PARIS 1959.

(2) BOUKBAKI N. Intégration ch IX. Hermann
PARIS 1969.

(3) NEVEU J. Bases mathématiques du calcul des probabilités
Masson PARIS 1964.

(4) FISZ M. Probability theory and mathematical statistics
S. Wiley 3è édition 1963.

(5) FALK et SOLAND An algorithm for separable non convex programming problems
II : non convex constraints '
Management science July 1971

(6) LIONS-TREMOLIERES-GLOWINSKI
Algorithmes d'optimisation.
I.R.I.A. Rocquencourt 1972.

(7) BENSOUSSAN-LIONS-TEMAM
Sur les méthodes de décomposition de décentralisation et de coordination et applications
I.R.I.A. Rocquencourt 1972

(8) MEDITCH J.S. Stochastic ptimal linear estimation and control
Mc GRAW-HILL series in electronic systems

(9) WONHAM W.M. On the separation theorem of stochastic control
SIAM J control N°2 1968.

(10) BENSOUSSAN A. Contrôle optimal stochastique de systèmes gouvernés par des équations aux dérivées partielles de type parabolique
I.R.I.A.

(11) JACOBSON et MAYNE Differential dynamic programming
American Elsevier company Inc. New York 1970

(12) SHREIDER The Monte Carlo method
Pergamon Press.

(13) TSE E. On dynamically allocating resources for stochastic non linear problems.
Systems control Inc. Palo Alto California.

(14) BELLMAN -DREYFUS
Applied dynamic programming
Princeton University Press 1962.

(15) LEMARECHAL -NEPOMIASTCHY -NISSEN
Méthodes numériques d'analyse de systèmes
Tome 1 Cahier IRIA N°9 Rocquencourt.

(16) POTIER D. Algorithmes de coordination.
Applications à la gestion d'unités de productions interdépendantes.
Thèse PARIS 1972.

(17) VIOT-QUADRAT Méthodes de simulations en programmation dynamique stochastique.
R.I.R.O. (à paraître).